Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Ing. Jiří Giesl

Kryptografický systém pro obrazy založený na deterministickém chaosu

Disertační práce

Studijní obor: Inženýrská informatika
Školitel: prof. Ing. Karel Vlček, CSc.

Zlín, Česká republika, 2010

Poděkování

Na tomto místě bych rád vřele poděkoval několika lidem, a to především mému školiteli prof. Ing. Karlu Vlčkovi, CSc., který mne provedl doktorským studiem a celou dobu mého působení na akademické půdě mne neustále podporoval.

Děkuji také svým kolegům v kanceláři, kteří vytvořili příjemné a přátelské prostředí, převážně Lukáši Tomšů a Ladislavu Běhalovi.

Největší díky patří mé rodině, hlavně bratru Petrovi, matce Lence a zesnulému otci Petrovi, bez kterých bych nikdy nemohl navštěvovat Univerzitu Tomáše Bati ve Zlíně. Celý život mi byli a jsou neskonalou oporou a netuším, kdy jim to vše budu moci oplatit. V neposlední řadě bych chtěl poděkovat také mé milující přítelkyni, Petře Štalmachové, která mne neustále žene dopředu.

"Chaos often breeds life, when order breeds habit."
Henry Adams

RESUMÉ

Tato disertační práce je zaměřena na návrh kryptografického systému pro šifrování obrazů pomocí nelineárních dynamických systémů, které vykazují chaotické chování. Chaotické systémy mají vlastnosti, které jsou velmi vhodné pro oblast kryptografie. Jedná se především o citlivost na počáteční podmínky a řídicí parametry a ergodicitu.

Většina chaotických šifer využívá chaotický systém pro generování předpisů a klíčů, které se následně používají pro samotný šifrovací proces. Navržená a prezentovaná metoda využívá jiný přístup. Složky obrazu zde hrají velmi důležitou roli pro samotné nastavení chaotického systému. Pozice a hodnoty pixelů jsou použity jako počáteční podmínky systému a za šifrovací klíče se považují samotné řídicí parametry. Protože lze obraz reprezentovat jako dvourozměrnou matici, používá se jako výchozí chaotický systém takový, který je popsán dvěma diskrétními iterativními mapami. Tyto dvě mapy jsou použity pro permutaci pixelů, tedy změnu jejich pozic. Pro šifrování barevných obrazů bylo nutné přidat třetí mapu, která má za úkol permutovat pixely mezi barevnými rovinami. To ovšem není z hlediska bezpečnosti dostačující, protože se v obraze stále nachází stejné množství informace. Chaotický systém tedy je rozšířen do čtyř-dimenzionální podoby, kde čtvrtá diskrétní mapa má za úkol modifikovat hodnoty pixelů. Protože je chaotický systém velmi citlivý, minimální odchylka v řídicích parametrech (šifrovacích/dešifrovacích klíčích) vede k naprosto rozdílnému chování a tedy i výstupu systému po určitém čase. To má za následek naprosto jiné rozložení pixelů v obrazu a velkou odlišnost zašifrovaných/dešifrovaných obrazů v případě použití rozdílných klíčů.

Experimenty ukazují, že navržená metoda má velmi silné dekorelační vlastnosti a je schopna vytvářet zašifrované formy obrazů s vysokou mírou entropie. To má za následek nečitelnost a nepředvídatelnost zašifrovaného obrazu. Řídicí parametry, které jsou považovány za klíče, mohou být oproti klasickým šifrovacím algoritmům reprezentovány jako reálná čísla. Prostor klíčů je tedy mnohem širší a šifra se tak stává rezistentní proti útoku hrubou silou, kdy se prochází celý prostor klíčů a hledá se správná hodnota klíče.

Kvůli vysoké výpočetní náročnosti šifrovacího algoritmu byla použita waveletová analýza, při které dochází k výběru signifikantních waveletových

koeficientů, které jsou zašifrovány. Výsledky ukazují předpokládané distorze dešifrovaných obrazů při stále velmi vysokém zabezpečení jejich zašifrovaných forem.

Kryptografický systém byl také podroben útoku evolučních algoritmů, kdy se diferenciální evoluce snažila o nalezení skrytého nastavení chaotického systému. Bylo dokázáno, že kryptografický systém je rezistentní i proti takovému typu heuristického útoku.

ABSTRACT

The main aim of this dissertation is to propose a cryptographic system for image encryption by means of nonlinear dynamical systems which exhibits chaos behaviour. Chaotic systems have several features which can be very useful in the field of cryptography. These features are especially sensitivity to initial conditions and control parameters and ergodicity.

Most encryption schemes of this type use chaotic system for the generation of rules or keys which are then used for the encryption process. Proposed and presented method is based on different principle. Components of image play very important roles for setting of chaotic system. Positions and values of pixels are considered as initial conditions of the system and encryption keys are represented as control parameters of that system. Because an image can by expressed as two-dimensional matrix, default chaotic system must be described by two discrete iterative maps. These two maps are utilized for permutation of pixel positions. For encrypting of colour images, it is necessary to add third discrete map because of permutation between colour planes. However, this permutation is not sufficient in terms of security, because of the same amount of information in that image. Therefore, chaotic system is extended to fourth-dimensional form where fourth discrete map is responsible for the modification of pixel values. Because of chaotic system sensitivity, even minimal divergence in control parameters (encryption/decryption keys) leads to different behaviour and outcome of that system after any time period. That is the reason of different distribution of pixels in an image and huge diversity of encrypted/decrypted images in the case of different keys.

Experiments show that proposed method has very strong de-correlation properties and that it is able to create encrypted forms of images with high value of entropy. As a result it will cause illegibility and unpredictability of encrypted image. Control parameters, which are considered as keys, can be represented as real numbers contrary of the classic cryptographic systems. Key space is then much larger and encryption scheme becomes resistant against the brute-force attack, when the right key is searched in the whole key space.

Cryptographic system is very expensive in terms of processing speed. Therefore, wavelet analysis was used for selection of significant wavelet

coefficients. These coefficients were then encrypted. Results show expected distortions in decrypted images while encrypted images are still very safe.

Proposed cryptographic system was also attacked by evolutionary algorithms, when differential evolution tried to reveal settings of chaotic system. It has been proven, that proposed system is resistant against this type of heuristic attack.

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM SYMBOLŮ A ZKRATEK

t	čas
n	diskrétní jednotka času
x	stavová proměnná
y	stavová proměnná
f'	derivace funkce f
λ	Ljapunovův exponent
Λ	vlastní číslo
P	matice obsahující obrazové složky
W	šířka obrazu
H	výška obrazu
D	počet barevných rovin obrazu
T	konstanta prahu
H	entropie
S	zdroj zpráv
E	střední hodnota
$r(d)$	křížová korelace v posunutí d
ψ	waveletová funkce
φ	měřítková funkce
s	měřítko waveletu

τ	poloha waveletu
$\downarrow 2$	podvzorkování dvěma
CF	účelová funkce
CFV	hodnota účelové funkce

ECB	Electronic Code Book
CBC	Cipher Block Chaining
CFB	Cipher-Feedback
OFB	Output-Feedback
CKBA	Chaotic Key-Based Algorithm
S-BOX	Substituční box
RGB	barevné roviny Red,Green,Blue
NPCR	Number of Pixel Change Rate
UACI	Unified Average Changing Intensity
Hi	High pass
Lo	Low pass
HH	High High pass
HL	High Low pass
LH	Low High pass
LL	Low Low pass

MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio
DERand1Bin	verze Diferenciální Evoluce
DERand2Bin	verze Diferenciální Evoluce
DEBest1Bin	verze Diferenciální Evoluce
DEBest2Bin	verze Diferenciální Evoluce

1. ÚVOD

Teorie chaosu získala své jméno ve druhé polovině 20. století a byla zkoumána zejména v oblasti matematiky a fyziky. Nicméně již v 19. století se objevily zmínky o jevech, které teorie chaosu popisuje. Za velmi významné zmínky se považují práce francouzského vědce Poincaré, který v jedné ze svých esejí, kde se zabýval Newtonovým gravitačním problémem Slunce, Měsíce a Země, konstatoval, že malá odchylka v počátečních podmínkách systému může vést po dlouhé době k naprosto rozdílným výsledkům. Teorie chaosu se tedy zabývá především nelineárními systémy, které v případě splnění určitých podmínek vykazují fenomén nazvaný chaos. Slovo chaos je odvozeno od řeckého „χαος“, které znamená „neuspořádanost“ a datuje se až do roku 800 př.n.l. [1]

Protože chování některých nelineárních systémů vypadá stochastiky, ale přesto dané systémy zůstávají deterministickými, můžou být využity v mnoha vědních disciplínách. Jednou z takových disciplín je i kryptografie. Vzhledem k tomu, že chaotické systémy jsou jednoduše implementovatelné a přesto vykazují citlivost na počáteční podmínky a ergodicitu, mohou být použity pro tvorbu kryptografického systému. Takový kryptografický systém má v případě správného návrhu velmi dobré difúzní a konfúzní vlastnosti při zabezpečení šifrovaných zpráv a silnou odolnost proti neoprávněnému čtení a některým druhům útoků.

Za poslední desetiletí bylo vyvinuto již poměrně velké množství šifer založených na deterministickém chaosu a to jak blokových, tak proudových šifer. Chaotický systém zde hraje velmi často roli generátoru, který vytváří předpisy (např. v podobě substitučních boxů) nebo klíče. Ty se následně starají o samotný šifrovací proces a případné maskování hodnot zprávy, která má být zašifrována. Pokud se nad tímto kryptografickým systémem zamyslíme, zjistíme zajímavý fakt. Je-li při šifrování různých zpráv nastaven chaotický systém stejně, pak vygeneruje i stejné předpisy nebo klíče a zprávy jsou tedy zašifrovány stejným způsobem. Pokud se tedy kryptoanalytikovi podaří odchytit zprávy, které byly zašifrovány jednoduchým systémem stejně, a u jedné zprávy komplikovanou analýzou odvodí šifrovací předpisy, pak tyto předpisy může aplikovat i na zprávy ostatní bez nutnosti další analýzy.

Tato disertační práce se zabývá návrhem kryptografického systému pro obrazy, který pro nastavení chaotického systému využívá i samotné složky obrazu. To má za následek vygenerování různých šifrovacích předpisů v případě použití stejných šifrovacích klíčů, ale rozdílných dat, která je potřeba zašifrovat.

Práce je rozčleněna do 9 hlavních kapitol:

První kapitola pouze uvádí tuto práci, zatímco již druhá kapitola vytyčuje základní cíle disertační práce.

Třetí kapitola pojednává o základních principech kryptologie a převážně se zabývá proudovými a blokovými kryptografickými systémy. Uvádí také, jaký je rozdíl mezi šifrováním obrazu a jiným typem dat.

Čtvrtá kapitola seznamuje s deterministickým chaosem a jeho základními vlastnostmi. Je zde popsáno několik chaotických systémů jak v diskrétní, tak spojité oblasti a v neposlední řadě i uvedena zmínka o podivných atraktorech.

Pátá kapitola uvádí použití chaosu v kryptografii. Jsou popsány některé výhody a nevýhody použití chaotických systémů v kryptografii a provedena rešerše kryptografických systémů založených na deterministickém chaosu.

Šestá kapitola je nejdůležitější kapitolou v této práci, protože se zabývá samotným návrhem kryptografického systému pro obrazy. Je zde uveden celý šifrovací proces a detailní analýza bezpečnosti zašifrovaných obrazů. Nechybí také srovnání s některými stávajícími chaotickými šiframi.

Sedmá kapitola popisuje využití waveletové transformace pro extrakci nejdůležitějších informací v obrazu, šifrování waveletových koeficientů a tím i urychlení šifrovacího procesu.

Osmá kapitola se zabývá nasazením diferenciální evoluce jako případné možnosti útoku na kryptografický systém.

Poslední kapitola sumarizuje některá fakta a diskutuje nad výsledky dosažených experimentů.

2. CÍLE DISERTAČNÍ PRÁCE

Hlavním cílem disertační práce bylo vyvinout kryptografický systém pro obrazy, který bude využívat nelineární dynamický systém vykazující chaotické chování. Tento kryptografický systém by měl využívat jiný způsob šifrování, splňovat všechny nároky kladené moderní kryptografií a v některých ohledech převyšovat stávající chaotické šifry.

Cíle disertační práce lze sumarizovat do následujících bodů:

- Navrhnout metodu pro šifrování obrazů
- Využít nelineární dynamický systém, který vykazuje chaotické chování
- Provést detailní analýzu zabezpečení
- Provést srovnání se stávajícími kryptografickými systémy
- Prozkoumat možnosti zavedení algoritmů do aplikací pracujících v reálném čase
- Ověřit rezistenci kryptografického systému proti útokům evolučních algoritmů

TEORETICKÁ ČÁST

3. KRYPTOGRAFICKÝ SYSTÉM

3.1. Kryptologie

Kryptologie je obor vědy zabývající se algoritmy zabezpečené komunikace. Mezi hlavní oblasti zájmu kryptologie patří vytváření metod, které mají zabezpečit přenos tajných informací a ochranu uložených dat; dále také zkoumání síly těchto algoritmů a metod pro jejich překonání. O kryptologii byl zájem většinou v souvislosti s vojenstvím nebo s diplomatickými záležitostmi a například v rané fázi se zabývá především utajenou písemnou komunikací. Díky vývoji komunikačních technologií dnes hraje zabezpečená komunikace velkou roli, především v obchodních, průmyslových a bankovních spojeních. Výzkum v oblasti kryptografických metod a hledání nových algoritmů stále pokračuje a je to dáno především díky vzrůstajícímu významu kryptologie v ekonomii.

Kryptologie zahrnuje dvě oblasti:

1. kryptografii
2. kryptoanalýzu

Kryptografie (šifrování) je proces transformace informací do nesrozumitelné formy tak, aby mohly být zaslány nezabezpečenou cestou, nebo mohly být uloženy v nezabezpečených souborech. Kryptografické postupy mohou být také použity pro osobní identifikace, digitální podpisy, kontroly přístupu apod.

Kryptoanalýza je věda, která se zabývá metodami, jak získat informace ze zašifrované zprávy, aniž by byla k dispozici tajná informace (většinou tajný klíč). Kryptoanalýza je tedy opakem kryptografie, přesto se používá i pro ověřování odolnosti kryptografického systému proti neoprávněnému dešifrování.

3.1. Kryptografický systém

Kryptografický systém je šifrovací algoritmus, který je obvykle znám, a jehož proces závisí na parametru nazývaném klíč. Pomocí šifrování můžeme přeměnit původní zprávu na formu, kterou člověk nemůže interpretovat, dokud nezná metodu a klíč použitý v tomto procesu. Dešifrování je inverzní proces, ve kterém zašifrované údaje jsou přepočteny na údaje původní. Jinými slovy,

základní myšlenka všech šifrovacích algoritmů je upravit zprávu tak, aby byl její obsah nesrozumitelný pro kohokoliv, kromě zamýšleného příjemce.

Diskrétní kryptografický systém lze charakterizovat jako uspořádanou pětici množin (P, C, K, E, D):

- množina původních zpráv P
- množina zašifrovaných zpráv C
- množina klíčů K
- množina šifrovacích a dešifrovacích transformací E a D

Zpráva $p \in P$ může být zastoupena konečnou posloupností symbolů zdroje zpráv. Proces šifrování $e(k, p) \in E$ lze považovat za funkci nebo algoritmus produkující zašifrovaný soubor dat $c \in C$. Symbol $k \in K$ označuje sadu parametrů a často se nazývá tajný klíč. Inverzní funkce k e je dešifrování $d(k, c) \in D$, které ze zašifrované zprávy c s tajným klíčem k, vyrábí zprávu původní podle (3.1). [2]

$$p = d(k, e(k, p)) \tag{3.1}$$

Pokud předpokládáme, že je útočníkovi známa struktura systému a že má přístup ke sdělovacímu kanálu, pak bezpečnost zašifrované zprávy závisí pouze na klíči k.

Šifrovací algoritmy mohou být klasifikovány do dvou základních skupin - blokové šifry a proudové šifry.

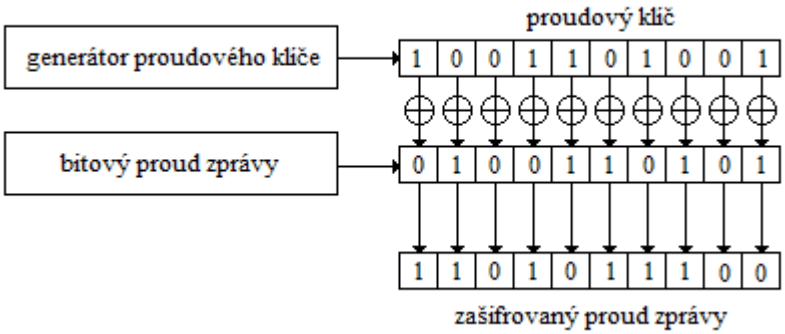
3.1.1. Proudová šifra

Proudová šifra je metoda, při které se každý znak zprávy šifruje zvlášť. Moderní proudové šifry využívají hlavní generátor, který vyrábí proud bitů $k_1, k_2 ..., k_n$ neboli bitový proud klíče. Pomocí něj se zašifruje původní zpráva $p_1, p_2 ..., p_n$ neboli bitový proud zprávy exkluzivním součtem XOR. Vznikne tak proud zašifrované zprávy $c_1, c_2 ..., c_n$ podle (3.2).

$$c_i = e(p_i) \tag{3.2}$$

kde $i = 1,2,..., n$.

Systém proudové šifry tak skryje původní zprávu změnou bitů náhodným způsobem. Útočník, který nezná klíč, nebude vědět, které bity se změnily (to odpovídá výskytu "1" na bitovém proudu klíče), nebo ty, které zůstávají beze změny ("0" na proudu klíče). Princip této binární aditivní proudové šifry je zobrazen na Obrázku 3.1.



Obrázek 3.1: Binární aditivní proudová šifra

Ideální proudová šifra by používala fyzický generátor náhodných čísel jako generátor klíče. Výstup tohoto generátoru ovšem nelze reprodukovat, proto dešifrování není možné, pokud se celý bitový proud klíče, který je stejně dlouhý jako bitový proud zprávy, nezašle příjemci. To je ovšem velmi nepraktické, proto se používají pseudonáhodné generátory čísel, které jsou řízeny relativně krátkými klíči. Používají se lineární souhlasné generátory, inverzní souhlasné generátory nebo zpětnovazební posuvné registry. Mezi nejčastěji používané proudové šifry patří RC4, FISH, Helix, SEAL nebo WAKE [3].

3.1.2. Bloková šifra

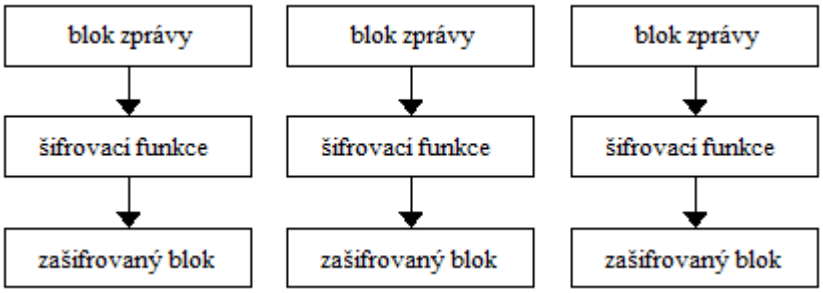
Na rozdíl od proudových šifer, kde je šifrován pouze jeden bit v jedné časové jednotce, se v případě blokových šifer šifrují celé bloky bitů současně. Mezi operační módy blokových šifer patří například „ECB (Electronic Code Book)“, kdy je každý blok šifrován zvlášť. Předpokládejme bloky zprávy $p_1, p_2 ..., p_n$ a šifrovací funkci e. Zašifrované bloky zpráv $c_1, c_2 ..., c_n$ lze vyjádřit podle (3.3).

$$c_i = e(p_i) \tag{3.3}$$

kde $i = 1,2,...,n$.

Principem módu ECB je vytvoření překladové tabulky (Code Book) na základě klíče. Indexem do této tabulky je blok původní zprávy a výstupem překladové tabulky je tedy blok výstupních dat, která odpovídají příslušnému

indexu. Využití tohoto módu ovšem není vhodné pro šifrování velkého množství dat, protože uvažujeme-li velikost bloku 64 bitů, pak by překladová tabulka měla 2^{64} položek, kde každá položka je velká 64 bitů. Výhodou může být případné paralelní šifrování bloků, protože bloky jsou na sobě nezávislé. Princip ECB módu je ukázán na Obrázku 3.2.



Obrázek 3.2: Bloková šifra ECB mód

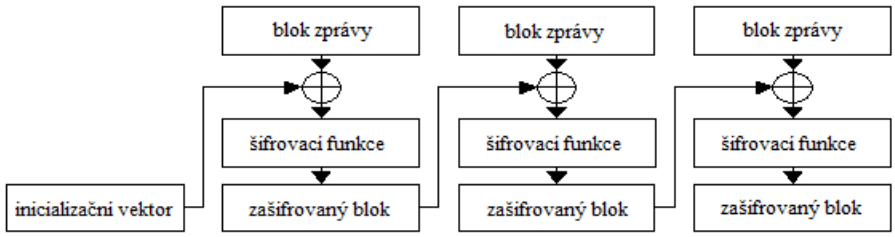
Existuje i druhý mód, tzv. "CBC (Cipher Block Chaining)", který zavádí zpětnou vazbu. Každý blok původní zprávy je nejprve modifikován předchozím blokem šifrované zprávy (prováděna XOR operace) a teprve poté se šifruje.

Předpokládejme bloky zprávy $p_1, p_2 ..., p_n$ a šifrovací funkci e. Zašifrované bloky zpráv $c_1, c_2 ..., c_n$ lze vyjádřit podle (3.4).

$$\begin{aligned} c_0 &= q \\ c_i &= e(p_i \oplus c_{i-1}) \end{aligned} \tag{3.4}$$

kde q je inicializační vektor a $i = 1,2,...,n$.

Bloky šifrované zprávy tedy závisí na všech předchozích blocích, které byly zpracovány. Takovýto způsob šifrování ovšem vyžaduje na počátku určitou hodnotu, která je označována jako inicializační vektor. Tento vektor by se měl zpravidla při každém šifrování měnit, aby nebylo možné získat určité statistiky při opakujících se prvních blocích zpráv. Pro generování inicializačního vektoru se používá zpravidla časový otisk neboli „timestamp". V případě, že se šifrují dvě totožné zprávy a inicializační vektory jsou na počátku různé, pak i zašifrované zprávy se budou lišit. Princip CBC módu je ukázán na Obrázku 3.3.



Obrázek 3.3: Bloková šifra CBC mód

Mezi další módy patří např. „CFB (Cipher-Feedback)“ nebo „OFB (Output-Feedback)“, které se chovají spíše jako proudové šifry s vlastní synchronizací, protože se šifrují velmi malé bloky dat. Používají se především v síťových aplikacích.

U blokových šifer zpravidla platí, že čím větší bloky se šifrují, tím větší je samotné zabezpečení zašifrované zprávy. Pokud by byly velikosti bloků malé, bylo by možné pro určitý klíč sestavit seznam vstupních a jim odpovídajících výstupních bloků. Šířka těchto bloků je tedy zpravidla 64 nebo 128 bitů. Mezi klasické blokové šifry patří algoritmy DES, IDEA, Blowfish, AES apod. [2]

3.1.3. Difúze a konfúze

Pro návrh blokových šifer se používají dva principy, které zajišťují rezistenci šifry proti statistickým metodám. Claude Shannon navrhl dvě metody - difúzi a konfúzi [4].

Cílem difúze je rozprostřít statistické charakteristiky původní zprávy do delších úseků zašifrované zprávy. Jestliže jeden znak původní zprávy ovlivňuje více znaků v zašifrované zprávě, potom se bigramové a vícegramové závislosti jazyka mohou projevit až v delších úsecích zašifrované zprávy. Čím větší je difúze původní zprávy do zašifrované zprávy, tím obtížnější je tyto závislosti zkoumat. Pro tyto účely se používá nejčastěji permutace – tzn. vzájemné prohazování jednotlivých složek původní zprávy.

Shannon dále definoval druhou metodu, jak znesnadnit statistickou kryptoanalýzu, tzv. konfúzi. Konfúze je metoda, jejímž cílem je učinit vztah mezi statistickými vlastnostmi zašifrované zprávy a klíčem co nejsložitější a zahrnující co největší části zašifrované zprávy a klíče. Jinými slovy - konfúze zajišťuje difúzi klíče do zašifrované zprávy tak, aby tato difúze byla složitá.

Nejjednodušším způsobem je substituce – tzn. náhrada jednotlivých bloků dat za bloky jiné.

Konfúzní a difúzní vlastnosti blokové šifry by měly být takové, že i když má útočník k dispozici mnoho příslušných dvojic bloků původní zprávy a bloků zašifrované zprávy (p_i, c_i), nemůže stále odvodit použitý šifrovací klíč nebo získat jinou užitečnou informaci, která by mu pomohla při kryptoanalýze.

3.1.4. Metody kryptoanalytických útoků

Rezistence proti neoprávněnému dešifrování je mírou výkonnosti kryptografického systému. Útoky na kryptografický systém lze rozdělit do čtyř skupin podle metody přístupu k informacím:

- útok typu **Ciphertext-only**: útočník má přístup ke sdělovacímu kanálu a může odchytit některé segmenty šifrované zprávy. Úkolem útočníka je pak odhalit původní zprávy a odvodit tajný klíč.
- útok typu **Known-plaintext**: k získaným zašifrovaným segmentům zná útočník také příslušné segmenty původní zprávy. Úkolem útočníka je pak odvodit tajný klíč.
- útok typu **Chosen-plaintext**: útočník nemá přístup pouze k zašifrovaným a původním segmentům, ale také si může zvolit původní zprávu, zašifrovat ji a získat tak příslušnou zprávu v zašifrované podobě pro potřeby srovnávání a analýz.
- útok typu **Chosen-ciphertext**: útočník si může zvolit různé segmenty zašifrované zprávy a následně získat příslušné segmenty původní zprávy.

Kromě výše zmíněných typických útoků existuje ještě jeden typ útoku nazvaný **Exhaustive key search**, jinak také nazýván jako **Brute-force attack**. Jde o procházení celého prostoru klíčů a hledání správného klíče pro dešifrování zprávy. Pokud je prostor klíčů příliš malý, tato metoda může podávat díky dnešní výpočetní kapacitě velmi dobré výsledky. [5]

Každý kryptografický systém, ať už založený na tradičních přístupech nebo na deterministickém chaosu, by měl splňovat obecné principy a být rezistentní proti výše zmíněným útokům.

3.1.5. Šifrování obrazu

Šifrování obrazu má oproti ostatním typům dat několik zvláštností, které je potřeba zmínit. Velmi důležitou vlastností je vysoká korelace hodnot sousedících pixelů v obrazu. Šifrovaný obraz musí tedy snížit tuto korelaci natolik, aby z něj nebylo možné vyčíst informaci o obrazu původním a nebylo možné statisticky determinovat hodnoty okolních pixelů. Distribuce pixelů ideálního šifrovaného obrazu by měla být rovnoměrná, aby bylo dosaženo maximální hodnoty entropie, a dále dva sousední pixely by měly být dekorelované.

Lidské vnímání zrakem je velmi robustní na degradaci nebo zašumění obrazu. Z tohoto důvodu lze také vybrat pouze některé signifikantní bity, které ovlivňují čitelnost obrazu, a s těmi pracovat. Konvenční šifrovací algoritmy pracují se všemi bity a je tedy vyžadována větší výpočetní síla pro zašifrování všech složek, přestože to není zapotřebí.

Signifikance určitých složek je velmi úzce spojena s kompresí obrazu. Komprese je v případě obrazů velmi vhodná, protože na rozdíl od textových informací jsou obrazy kapacitně velmi rozsáhlé. Pokud se bere v potaz 24-bitový obraz o velikosti 256x256 pixelů, pak tento obraz obsahuje 192kB dat. Pro potřeby aplikací reálného času je toto množství dat nevhodné. Ve většině případů je tedy komprese provedena před samotným procesem šifrování, aby se získalo menší množství dat, které je nutné zašifrovat.

Další vlastností obrazu je nižší sensitivita. Pokud u textové informace změníme u všech symbolů pouze 1 bit, zapříčiní to větší změnu informace než v případě, kdy tuto změnu provedeme u všech pixelů v obrazu. I z tohoto důvodu je tedy nutné při šifrování provést tak velké změny, že to zapříčiní jeho celkovou nečitelnost.

Při návrhu vhodného kryptografického systému by tedy měly být brány v potaz všechny odlišnosti obrazu od jiných typů dat.

4. DETERMINISTICKÝ CHAOS

4.1. Chaotické systémy

Mnoho oblastí vědy a výzkumu předpokládá možnost predikce a opakování experimentů. Nicméně byly nalezeny velmi jednoduché deterministické systémy, které nelze predikovat. Teorie chaosu patří do pole nelineární dynamiky, která je součástí oblasti dynamických systémů. Hlavní vlastností systémů, které vykazují chaotické chování, je velká citlivost na počáteční podmínky a řídicí parametry. Zvolíme-li dva nekonečně blízké body, které budou reprezentovat počáteční podmínky systému, pak tyto dva body se budou při běhu systému od sebe exponenciálně vzdalovat. Budoucí stav systému není možné žádným způsobem předpovědět.

Příkladem může být počasí. Přestože se atmosféra podřizuje deterministickým fyzikálním zákonům, nelze počasí předpovědět v delším časovém horizontu, protože počasí vykazuje extrémní citlivost na počáteční podmínky. Velmi malá změna zapříčiní velké změny, které ovlivní počasí ještě více v následující dny. Tento jev se populárně nazývá „motýlí efekt“, podle kterého například mávnutí motýlích křídel v Brazílii může vyvolat tornádo v Texasu. Protože počáteční podmínky nikdy nebudeme s absolutní přesností znát, pak ani dlouhodobé předpovědi nejsou možné, přestože jsou nám fyzikální zákony známy a chovají se deterministicky.

Dynamické systémy jsou velmi často popsány diskrétními předpisy nebo diferenciálními rovnicemi, které reprezentují chování systému po krátké časové období.

Předpokládejme diskrétní dynamický systém zapsaný v jednoduché formě (4.1)

$$x_{n+1} = f(x_n), \qquad f : I \to I, \qquad x_0 \in I \tag{4.1}$$

kde x_n je skalár nebo vektor a f je spojitá nelineární funkce na intervalu I. Pro deterministický chaos je nelinearita funkcí nutná, nikoliv však postačující podmínka, protože mnoho nelineárních funkcí chaos negeneruje. O nelineárním dynamickém systému můžeme říci, že je chaotický, pokud splňuje následující podmínky: [6]

1. citlivost na počáteční podmínky

$$\exists \delta > 0 \quad \forall x_0 \in I, \varepsilon > 0 \quad \exists n \in N, y_0 \in I : \\ |x_0 - y_0| < \varepsilon \Rightarrow \left| f^n(x_0) - f^n(y_0) \right| > \delta \tag{4.2}$$

2. topologická tranzitivita

$$\forall I_1, I_2 \subset I, \exists x_0 \in I_1, n \in N : \\ f^n(x_0) \in I_2 \tag{4.3}$$

3. hustota periodických bodů P v I

$$P = \{p \in I \mid \exists n \in N : f^n(p) = p\} \\ \overline{P} = I \tag{4.4}$$

kde f^n je n-tá iterace funkce f.

Z výše uvedených vlastností lze vidět, že při mírně odlišných počátečních podmínkách jsou výstupy systému po několika iteracích diametrálně odlišné. Topologická tranzitivita zajišťuje ergodicitu, což znamená, že pokud rozdělíme stavový prostor do konečného počtu oblastí, pak každá orbita systému projde všemi těmito oblastmi.

4.1.1. Diskrétní systémy

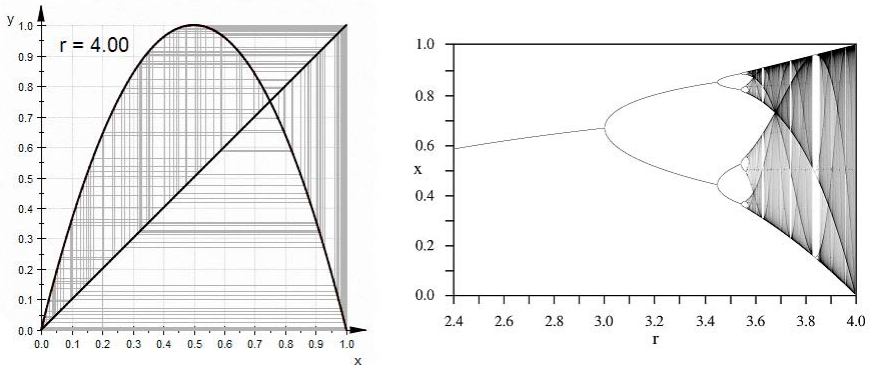
Logistická mapa

Logistická mapa je jedním z nejjednodušších dynamických nelineárních systémů, který vykazuje chaotické chování. Původně byla vytvořena Pierrem Francoisem Verhulstem jako demografický model, kdy populační růst v jednom časovém období závisí na jeho růstu v předchozím období. Jakmile celková část populace dosáhne určité hodnoty, která značí příliš mnoho organismů v uzavřeném prostoru, pak začne populační růst klesat. A naopak v menší populaci dochází k jejímu růstu. Výpočet růstu populace vychází z velmi jednoduchého matematického modelu (4.5).

$$x_{n+1} = r x_n (1 - x_n) \tag{4.5}$$

kde $x_n \in (0,1)$ reprezentuje počet jedinců (dolní hranice 0 znamená vymření populace, horní hranice 1 znamená plný stav) v n-té generaci a parametr $r \in (0,4)$ je velikost růstu. Na Obrázku 4.1 je zobrazen atraktor a bifurkační

diagram logistické mapy, který zachycuje hodnoty parametru r kdy dochází ke zdvojování počtu orbit.



Obrázek 4.1: Atraktor a bifurkační diagram logistické mapy

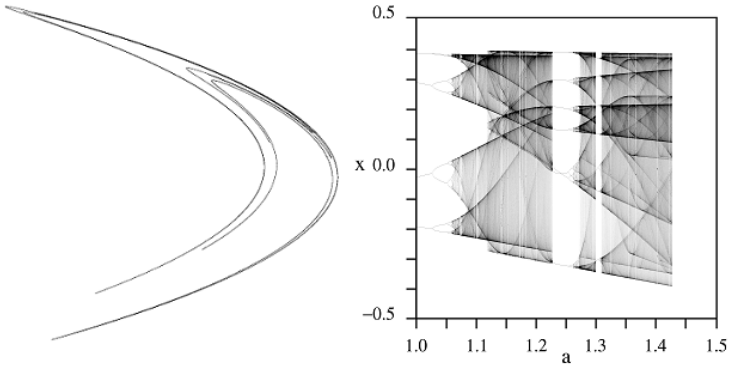
Z bifurkačního diagramu lze vidět, že pro míru růstu menší než 300% (parametr $r < 3.0$) se po určité době stav populace ustálí na určité hodnotě a v dalších generacích (iteracích) již nenastávají žádné další výrazné změny. Jestliže je míra růstu mezi 300% až 345% ($3.45 < r < 3.0$), pak systém začne oscilovat mezi dvěma stavy – každé časové období se tedy střídají dvě různé velikosti populace. S dalším zvyšováním míry růstu se zvětšuje i počet stavů a systém osciluje mezi čím dál větším množství stavů. Při míře růstu větším než 357% ($r > 3.57$) se chování systému již stává chaotickým a nelze pro další časová období předpovědět, jak se velikost populace bude vyvíjet, protože se nikdy neustálí na konstantní hodnotě.

Hénonova mapa

Hénonova mapa patří mezi nejjednodušší dvou-dimenzionální diskrétní mapy. Iterativní mapy (4.6) odvodil Michel Hénon při studiu pohybu astronomických objektů.

$$\begin{aligned} x_{n+1} &= 1 + y_n - a x_n^2 \\ y_{n+1} &= b x_n \end{aligned} \tag{4.6}$$

Lze vidět, že se jedná o rozšíření jedno-dimenzionální kvadratické mapy, které závisí na dvou řídicích parametrech a a b. Atraktor Hénonovy mapy pro parametry $a = 1.4$ a $b = 0.3$ a jeho příslušný bifurkační diagram je zobrazen na Obrázku 4.2.



Obrázek 4.2: Atraktor a bifurkační diagram logistické mapy

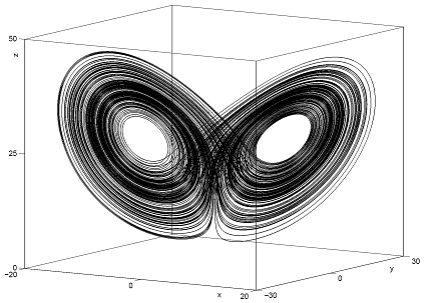
4.1.2. Spojité systémy

Lorenzův atraktor

Lorenzův atraktor je prvním prozkoumaným dynamickým systémem s tzv. podivným atraktorem. Tento systém byl objevem Edwardem Nortonem Lorenzem v roce 1963 při modelování počasí. Vzhledem k tehdejší výpočetní náročnosti byl model zjednodušen na 3 diferenciální rovnice (4.7).

$$\begin{aligned} \frac{dx}{dt} &= -ax + ay \\ \frac{dy}{dt} &= bx - y - zx \\ \frac{dz}{dt} &= -cz + xy \end{aligned} \tag{4.7}$$

Na Obrázku 4.3 lze vidět, že orbita Lorenzova atraktoru je neperiodická a dochází zde pouze k protínání drah, ne k jejich splynutí.



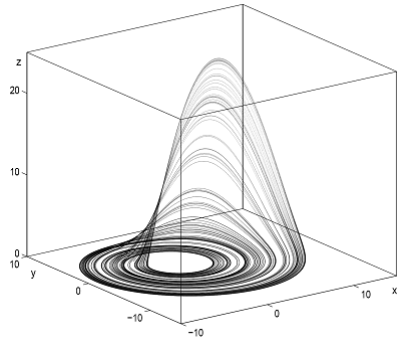
Obrázek 4.3: Lorenzův atraktor

Rösslerův atraktor

Rösslerův atraktor je uměle vytvořený systém generující podivný atraktor. Byl představen v roce 1976 a vychází z Lorenzova atraktoru. Oproti němu generuje ovšem pouze jedno křídlo. Rösslerův atraktor je dán 3 diferenciálními rovnicemi (4.8).

$$\begin{aligned} \frac{dx}{dt} &= -y - z \\ \frac{dy}{dt} &= x + ay \\ \frac{dz}{dt} &= b + xz - cz \end{aligned} \tag{4.8}$$

Na Obrázku 4.3 lze vidět, že první dvě rovnice, které jsou lineární, vyvolávají oscilaci kolem proměnných x a y . Se zvyšováním řídicí proměnné a dochází k zesilování oscilací. Přitom pohyb v x a y je závislý na nelineárním vyjádření v z .



Obrázek 4.4: Rösslerův traktor

4.2. Ljapunovův exponent

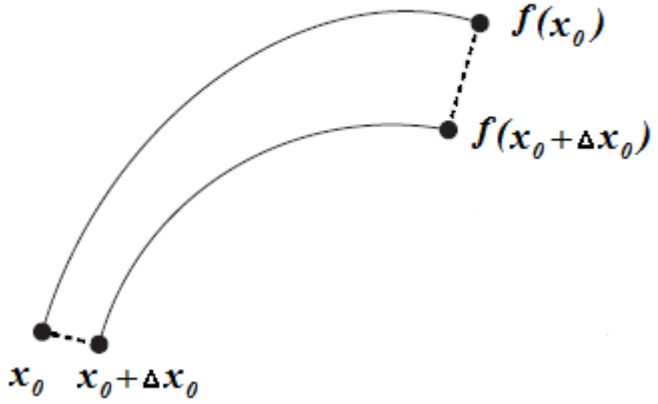
Citlivost na počáteční podmínky u chaotických systémů je kvantifikována parametrem, který se nazývá Ljapunovův exponent. Ljapunovův exponent vyjadřuje, zda blízké dráhy systému konvergují nebo divergují. Pro každou dimenzi systému existuje právě jeden Ljapunovův exponent. Lze ovšem říci, že maximální exponent nejvíce ovlivňuje dlouhodobé chování systému.

Předpokládejme diskrétní systém uvedený ve formě (4.1). Dále mějme dva velmi blízké počáteční body x_0 a $x_0 + \Delta x_0$. Po jedné iteraci diskrétního systému (4.1) jsou body odděleny podle (4.9)

$$\Delta x_1 = f(x_0 + \Delta x_0) - f(x_0) \cong \Delta x_0 f'(x_0) \tag{4.9}$$

kde f' je $\frac{df}{dx}$. [1]

Vše lze jednoduše vyjádřit pomocí Obrázku 4.5.



Obrázek 4.5: Separace počátečních podmínek

Lokální Ljapunovovo číslo, které lze chápat jako míru rozpínání v bodě x_0, je definováno jako (4.10)

$$e^{\lambda_i} = \left| \frac{\Delta x_1}{\Delta x_0} \right| \tag{4.10}$$

Samotný lokální Ljapunovův exponent v bodě x_0 lze tedy zapsat podle (4.11)

$$\lambda_i = \ln e^{\lambda_i} \cong \ln \left| f'(x_0) \right| \tag{4.11}$$

Globální Ljapunovův exponent (4.12) je střední hodnotou lokálních Ljapunovových exponentů (4.11) přes velké množství iterací. [1]

$$\lambda = \lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln \left| f'(x_n) \right| \tag{4.12}$$

Počítání lokálních Ljapunovových exponentů je velmi blízké určování vlastních čísel. Máme-li vlastní číslo Λ_i v bodě x_0, pak lokální Ljapunovovo číslo je absolutní hodnotou vlastního čísla a lokální Ljapunovův exponent lze tedy vyjádřit jako (4.13)

$$\lambda_i = \ln\left|\Lambda_i\right| \tag{4.13}$$

Rozdíl mezi vlastním číslem a globálním Ljapunovovým exponentem je zřejmý. Vlastní číslo je komplexní číslo, které vyjadřuje míru změny v daném bodě. Naproti tomu globální Ljapunovův exponent je reálné číslo vyjadřující průměrnou změnu vývoje celého systému. Systém s n dimenzemi má n globálních Ljapunovových exponentů a n vlastních čísel v každém bodě [1].

Globální Ljapunovův exponent tedy vyjadřuje průměrnou míru separace dvou počátečních hodnot nebo také průměrné roztažení prostoru. Kladná hodnota exponentu značí divergenci počátečních hodnot, záporná hodnota jejich konvergenci. Chaotický systém musí mít aspoň jeden Ljapunovův exponent kladný, tedy aspoň v jednom směru se musí sousední trajektorie od sebe exponenciálně vzdalovat.

4.3. Podivný atraktor

Chaotický pohyb lze vykreslit v abstraktním prostoru, který je nazýván „fázový prostor“. V něm každá osa představuje jednu dimenzi stavu a čas je zde implicitní. V průběhu vyvíjení systému vzniká ve fázovém prostoru křivka a po nějaké době začne tato křivka zvýrazňovat strukturu, které se říká „atraktor“. Atraktor je množina stavů, ke kterým ostatní stavy systému po určité době konvergují, a lze tedy říci, že atraktor je konečným stavem systému.

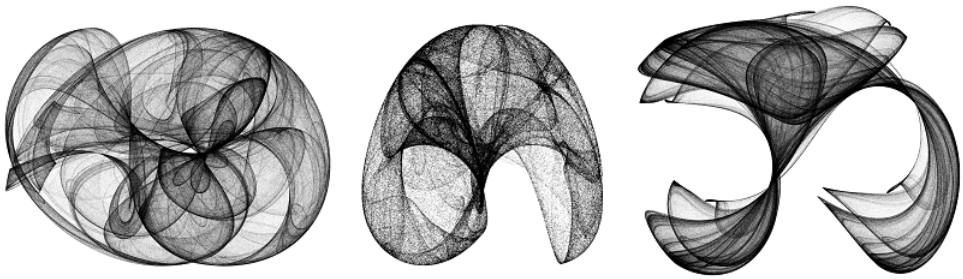
Obecně existuje 5 základních typů atraktorů: „bodový“, „periodický“, „kvazi-periodický“, „chaotický“ a „podivný“.

Podivné traktory mohou být vytvořeny několika způsoby, např. kvadratickými (4.14) nebo trigonometrickými (4.15) mapami. Řídicí parametry a,b,c,d,e,f,g,h,i,j,k,l definují chování chaotického systému [7].

$$\begin{aligned} x_{n+1} &= a + b\cdot x_n + c\cdot x_n^{\,2} + d\cdot x_n y_n + e\cdot y_n + f\cdot y_n^{\,2} \\ y_{n+1} &= g + h\cdot x_n + i\cdot x_n^{\,2} + j\cdot x_n y_n + k\cdot y_n + l\cdot y_n^{\,2} \end{aligned} \tag{4.14}$$

$$x_{n+1} = a \cdot \sin(b \cdot y_n) + c \cdot \cos(d \cdot x_n)$$
$$y_{n+1} = e \cdot \sin(f \cdot y_n) + g \cdot \cos(h \cdot x_n) \tag{4.15}$$

Na Obrázku 4.6 lze vidět příklad některých podivných atraktorů.



Obrázek 4.6: Podivné atraktory

Zobecněná definice atraktoru neexistuje, přesto můžeme vytyčit jeho základní vlastnosti: [7]

- jedná se o omega množinu, což je limitní množina bodů i v případě, že čas jde do nekonečna
- jedná se o invariantní množinu bodů
- je ohraničený a nerozpíná se do nekonečna
- má fraktální strukturu - má neceločíselnou dimenzi a je sobě-podobný
- má dostatečnou hustotu period - každý bod na atraktoru je velmi blízký k aspoň jedné z orbit, většina orbit mají extrémně dlouhou periodu
- je tranzitivní - z každého bodu na atraktoru nás dynamika provede po velmi blízkých vzdálenostech ke všem bodům na tomto atraktoru
- je ergodický - není tedy složen z menších atraktorů, které by se k sobě blížily nebo dotýkaly
- je strukturálně stabilní, zatímco fixní body atraktoru jsou nestabilní
- je chaotický - dvě blízké počáteční podmínky se od sebe exponenciálně oddalují

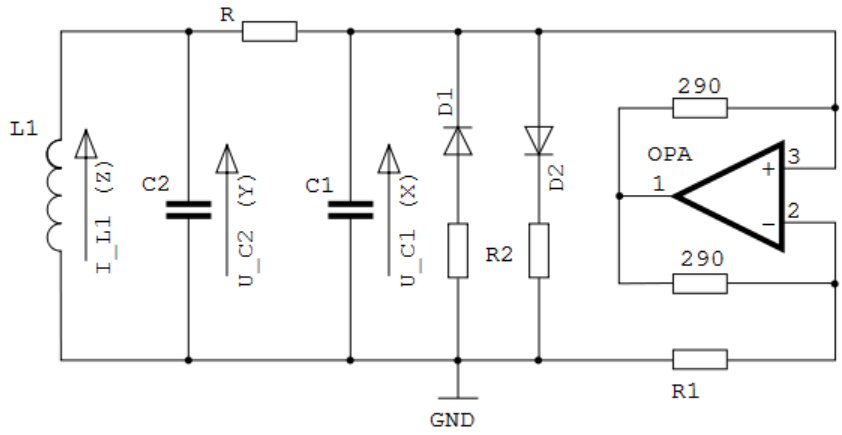
5. VYUŽITÍ CHAOSU V KRYPTOGRAFII

5.1. Výhody použití chaotických systémů

Vzhledem k povaze chaotických funkcí, nemají klasické metody kryptoanalýzy smysl. Klasická kryptoanalýza používá různé metody, jako jsou statistické analýzy, vyčerpávající průzkum hrubou silou a využívání slabin v šifrovacím algoritmu. Tyto metody ovšem nelze použít na kryptografické systémy založené na deterministickém chaosu hlavně kvůli tomu, že chaotické systémy vykazují stochastické chování a jsou tedy imunní vůči statistickým analýzám. Útok hrubou silou zde také nenajde uplatnění, protože klíče jsou reálného charakteru a prohledávat spojitý rozsah čísel je nesmírně náročné. V tomto se velmi liší od tradičních šifrovacích metod, kde byly využívány klíče z velké, ale konečné a diskrétní oblasti.

Tradiční kryptografické systémy používají algoritmy, které s lineárním počtem iterací nebo délkou klíče zvyšují difúzi a konfúzi pouze lineárním způsobem. Naproti tomu chaotické systémy v této oblasti opět ty klasické překonávají.

Další výhodou chaotického šifrování je skutečnost, že může být přímo hardwarově implementováno, bez toho aniž by bylo potřeba použít DAC (digital to analog converter). Jakákoliv forma konverze představuje určitou ztrátu přesnosti. Pokud je zpráva reprezentována ve své zašifrované formě, je potřeba zajistit co největší přesnost. Fyzické provedení chaotické funkce může být postaveno na hardwarovém obvodu Van der Polova oscilátoru nebo Chuova oscilátoru [10]. Zapojení Chuova oscilátoru je zobrazeno na Obrázku 5.1. Tímto způsobem je možné implementovat kryptografický systém, který není omezen současnou výpočetní technikou a její rychlostí. Takový obvod je také schopen pracovat se spojitě-analogovými signály v plné rychlosti.



Obrázek 5.1: Chuův oscilátor

Lze tedy konstatovat, že základní výhody chaotického šifrování jsou odolnost vůči tradičním formám útoků, jednoduchost při zavádění různorodosti algoritmů a vhodnost realizace pro analogové systémy [8].

5.2. Nevýhody použití chaotických systémů

Každá metoda má i své nevýhody a kryptografie pomocí chaosu není výjimkou. Jedna z výhod, kdy je chaotické šifrování odolné proti konvenční kryptoanalýze, se ukazuje také jako nevýhoda. Vzhledem k obtížnosti kryptoanalýzy nelze bezpečnost systému snadno kvantifikovat a tedy i úroveň bezpečnosti není dosud dobře charakterizována. Klasický kryptografický algoritmus jakým je algoritmus RSA (který v roce 1978 vynalezl Ron Rivest, Adi Shamir a Leonard Adleman) měl svoji bezpečnost založenou na skutečnosti, že je velmi obtížné rozložit velmi velké číslo na součin prvočísel. Z čísla $n = pq$ je tedy v rozumném čase prakticky nemožné zjistit činitele p a q, neboť není znám žádný algoritmus faktorizace, který by pracoval v polynomiálním čase vůči velikosti binárního zápisu čísla n [9]. Žádná podobná skutečnost u chaotických systémů neexistuje.

Za další slabinu lze považovat fakt, že chaotický systém není bezpečný pro šifrování velmi dlouhých zpráv. Můžeme tuto slabost přisoudit vlastnosti systému, že chaotické mapování se může opakovat nebo jít do určité orbity pro různé počáteční podmínky. Sledováním výstupů šifrovacího algoritmu, který je spuštěn se stejnými parametry pro mnoho různých zpráv, je možné

rekonstruovat část nebo všechna chaotická mapování tak, že sestavíme všechny možné orbity odpozorované ze šifrovaného výstupu. Jde tedy o pokus obnovit fázový prostor systému vytvořením *n*-tého vzorku proti (*n+1*)-mu vzorku šifrované zprávy [8].

V praktických implementacích chaotických systémů vzniká také několik různých problémů – jak v digitálních, tak v analogových oblastech.

V analogové oblasti je problém se šumem. Šum je nedílnou součástí ve všech systémech, jejichž teplota okolí je vyšší než absolutní nula a tento fenomén je tedy nevyhnutelný. Vyrobit dva identické a synchronizované chaotické systémy na hardwarové úrovni je opravdu velmi obtížné, pokud zde existuje šum, který neustále zapříčiňuje odchylky ve výstupech systémů a tedy nemožnosti plné synchronizace. Synchronizace dvou generátorů je v kryptografii zapotřebí a může se stát slabým místem celého kryptografického systému.

Problémy u digitálních systémů jsou ještě větší. Pokud jde o výběr klíče pro šifrování zprávy, musíme tyto klíče omezit pouze na konečný počet. To znamená, že namísto nekonečné a spojité oblasti čísel jsme limitováni na její podmnožinu, která může zapříčinit periodicitu namísto chaotického chování [8].

Další související problém je, že čísla v plovoucí řádové čárce mohou být na různých hardwarových platformách reprezentována jinak. Čísla s plovoucí řádovou čárkou jsou reprezentována jako reálná čísla (mantisa) a pevný počet číslic reprezentující exponent (obvykle o základu 10). Přestože existují standardy týkající se reprezentace tohoto datového typu podle IEEE, různé mikroprocesory mají různé datové velikosti od 4 bitů do 128 bitů, takže je nemožné plně standardizovat způsob počítání čísel.

Lze tedy vidět některé nevýhody chaotických systémů, jako je šifrování dlouhých zpráv a problémy s praktickou implementací.

5.3. Současný stav chaotických šifer obrazu

Chaotické systémy byly v oblasti kryptografie využity nejrůznějšími způsoby. V práci [11] byla odvozena jedno-dimenzionální mapa, která vykazovala chaotické chování na určitém intervalu hodnot. Tato mapa byla použita pro generování sekvence pseudonáhodných čísel, které byly použity pro šifrování zprávy. Tato práce byla ovšem v [12] zkritizována, protože

navržená metoda nebyla vhodná pro digitální výpočetní techniku. Po diskretizaci použité mapy produkoval generátor cykly, jejichž délka byla velmi krátká.

Práce [13] používala logistickou mapu pro generování sekvence čísel s plovoucí řádovou čárou. Tato čísla byla následně konvertována do binární sekvence, která byla XOR-ována s bity šifrované zprávy. Konverze na binární sekvenci byla provedena na základě výběru jednoho ze dvou intervalů, které reprezentovaly hodnoty 0 a 1. Řídicí parametr logistické mapy spolu s počáteční podmínkou byly brány jako šifrovací klíče.

Objevilo se i použití několika různých chaotických map zároveň [14]. Použitím *m* map lze získat šifrovací klíč. Po několika iteracích map lze z jejich výstupů extrahovat *m* bytů, které jsou následně zkombinovány XOR operací. Tento proces je opakován do té doby, dokud není vytvořeno jednorázové heslo (one-time pad). Toto heslo je poté XOR-ováno se šifrovanou zprávou.

O generování jednorázového hesla se pokoušeli i jiní v práci [15]. Zde byla použita dvou-dimenzionální matice o *NxM* blocích, která se vzdáleně podobala zobecněné mapě „baker“. Šifrovací klíč byl použit ke generování bloku bitů, který byl opakovaně permutován a XOR-ován sám se sebou. Po $log_2(NxM)$ iteracích byl získán blok, který vypadal pseudonáhodně a který byl použit pro XOR operaci se šifrovanou zprávou.

Práce [16] popisuje šifrovací systém založený na jedno-dimenzionálním buněčném automatu. Zpráva, která byla šifrována, mohla vést náhodně k několika různým zašifrovaným formám. Tyto zašifrované formy ovšem byly větší než původní zpráva.

Šifra obrazu představená v [17] využívala dvou-dimenzionální „baker“ mapu pro účely permutace pixelů. Tato mapa byla iterativně aplikována na obraz a ukázalo se, že permutace vyvolané „baker“ mapou vykazují nahodilost. Mapa byla i dále rozšířena na třetí dimenzi a zakomponován jednoduchý difúzní mechanismus.

V [18] byla představena metoda nazvaná CKBA (Chaotic Key-Based Algorithm), která generovala časovou řadu na základě chaotické mapy a z této řady byla následně vytvořena binární sekvence, která byla považována za tajný klíč. Na základě binární sekvence byly pixely obrazu přeskupeny a hodnoty pixelů byly pozměněny na základě XOR nebo XNOR operace s tajným

klíčem. Tato metoda byla velmi jednoduchá, ovšem málo rezistentní proti útokům typu chosen/known-plaintext a také proti útoku hrubou silou.

Efektivní generátor byl představen v práci [19], kde byly použity tři paralelně spojené logistické mapy a tři kvantizační jednotky pro tvorbu binární sekvence. Tato sekvence byla opět použita pro šifrování obrazu. Šifra byla rezistentní proti různým útokům, jakým je i například útok neuronovou sítí, rekonstrukce klíče pomocí reverzních iterací logistické mapy a také proti útoku typu known-plaintext.

V pracích [20, 21] byla navržena šifra obrazu založená na 3D mapě typu „cat“ a „baker“; ta vytvářela 3D matici, která se následně postarala o difúzi pixelů. Nicméně v [22] bylo dokázáno, že tyto šifry mají velmi malý prostor klíčů. Proto bylo navrženo použití standardní mapy pro konfúzi, zatímco logistická mapa se starala o difúzi pixelů. Tento proces byl ovšem výpočetně velmi komplexní, proto byl v [23] navržen způsob, jak šifrování urychlit. Protože je difúze náročnější než konfúze, zlepšení spočívalo ve snížení počtu rund difúzí a explicitní zakomponování velmi jednoduché modifikace hodnot pixelů do konfúzního procesu. Výsledkem byla podobná úroveň zabezpečení při sníženém počtu rund šifrovacího procesu a tedy při snížené komplexnosti šifrovacího algoritmu.

Objevily se i některé další typy šifer, které využívají chaotické ergodické matice pro permutaci pixelů a algebraické operace S-boxů pro difúzi [24]. Lze tak získat velmi dobré statistické vlastnosti zašifrovaného obrazu a rezistenci proti různým statistickým útokům.

Většina šifer obrazu je založena na blokovém schématu, pouze některé jsou implementovány jako proudové šifry [25]. Výhody postupného zpracování každé složky obrazu se ovšem negativně projevují na výsledném zabezpečení zašifrované formy. Taková šifra má většinou slabé dekorelační vlastnosti a zanechávají určitou statistickou závislost mezi pixely.

PRAKTICKÁ ČÁST

6. KRYPTOGRAFICKÝ SYSTÉM PRO OBRAZY

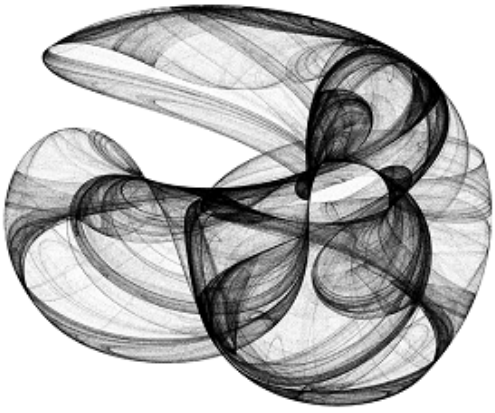
6.1. Návrh kryptografického systému

Většina šifer obrazu, které jsou založeny na principu deterministického chaosu, vytváří substituční boxy pro úpravy hodnot pixelů nebo generují bitový proud klíče, který lze použít pro šifrovací účely. Obraz je tedy maskován hodnotami, které byly vytvořeny pomocí systému vykazujícím chaotické chování. V dostupné literatuře ovšem mnoho zmínek o naprosto jiných přístupech. Hodnoty pixelů obrazu mohou být použity přímo jako výchozí nastavení chaotického systému. Pokud hodnotu pixelu nastavíme jako počáteční podmínku systému, po několika iteracích získáme hodnotu novou, kterou můžeme použít pro případné šifrování. Tento způsob se liší od jiných chaotických šifer, u kterých vždy stejné šifrovací klíče vedou ke generování stejných šifrovacích předpisů. Vzhledem k tomu, že v navržené metodě i pixely hrají roli při nastavení systému, šifrovací předpisy při stejných šifrovacích klíčích budou pro různé obrazy vždy jiné.

Navržený kryptografický systém využívá vlastností chaotického systému, který generuje podivný atraktor. V této práci je jako výchozí chaotický systém použit Cliffordův systém, který je popsán dvěma na sobě závislými iterativními mapami (6.1).

$$\begin{aligned} x_{n+1} &= \sin(a \cdot y_n) + c \cdot \cos(a \cdot x_n) \\ y_{n+1} &= \sin(b \cdot y_n) + d \cdot \cos(b \cdot x_n) \end{aligned} \tag{6.1}$$

Výše uvedený předpis je velmi podobný předpisu trigonometrických map (4.15). Na rozdíl od něj ovšem obsahuje menší počet řídicích parametrů, které jsou různé od hodnoty 1. Obrázek 6.1 zobrazuje atraktor Cliffordova systému pro před-definované řídicí parametry.



Obrázek 6.1: Cliffordův atraktor

Z (6.1) lze vidět, že Cliffordův systém je vytvořen jako dvou-dimenzionální systém. Pro účely šifrovacího algoritmu byl jeho předpis rozšířen do dalších dvou dimenzí. Zmodifikovaný předpis čtyř-dimenzionálního Cliffordova systému tedy vypadá podle (6.2).

$$\begin{aligned} x_{n+1} &= \sin(a \cdot y_n) + c \cdot \cos(a \cdot x_n) \\ y_{n+1} &= \sin(b \cdot x_n) + d \cdot \cos(b \cdot y_n) \\ z_{n+1} &= \sin(e \cdot x_n) + f \cdot \cos(e \cdot z_n) \\ w_{n+1} &= \sin(g \cdot y_n) + h \cdot \cos(e \cdot w_n) \end{aligned} \tag{6.2}$$

kde $a, b, c, d, e, f, g, h \in R$ jsou řídicí parametry chaotického systému, x_n, y_n, z_n a w_n jsou výstupy systému v n -té iteraci. Základním principem navrženého algoritmu je využívat Cliffordův systém pro šifrování každého pixelu zvlášť.

Uvažujme tří-rozměrnou matici P, která reprezentuje obraz. Matice P obsahuje hodnoty pixelů $p_{i,j,k} \in P$ daného obrazu, kde $i = 0,1,2,...,W$, $j = 0,1,2,...,H$ a $k = 0,1,2,...,D$. Rozměry W, H a D určují šířku, výšku a hloubku matice P. Hloubku D si lze představit jako počet barevných složek obrazu. V této práci se pracuje převážně s tří-barevným modelem RGB.

Definujme $p_{i,j,k}$ jako pixel na souřadnicích (i, j, k) v matici P. Pozice i, j, k a hodnota $p_{i,j,k}$ jsou vloženy do (6.2) jako počáteční podmínky příslušné mapy podle (6.3).

$$x_0 = i, \quad y_0 = j, \quad z_0 = k, \quad w_0 = p_{i,j,k} \tag{6.3}$$

Po m-té iteraci rozšířeného Cliffordova systému (6.2) jsou k dispozici výstupy x_m, y_m, z_m a w_m. Na tyto hodnoty je aplikována kvantizační jednotka, která má za úkol zajistit, aby všechny hodnoty byly v povolených hranicích. Předpis této kvantizační jednotky lze definovat podle (6.4)

$$output = (input \cdot U)(\mathrm{mod}\, T) \tag{6.4}$$

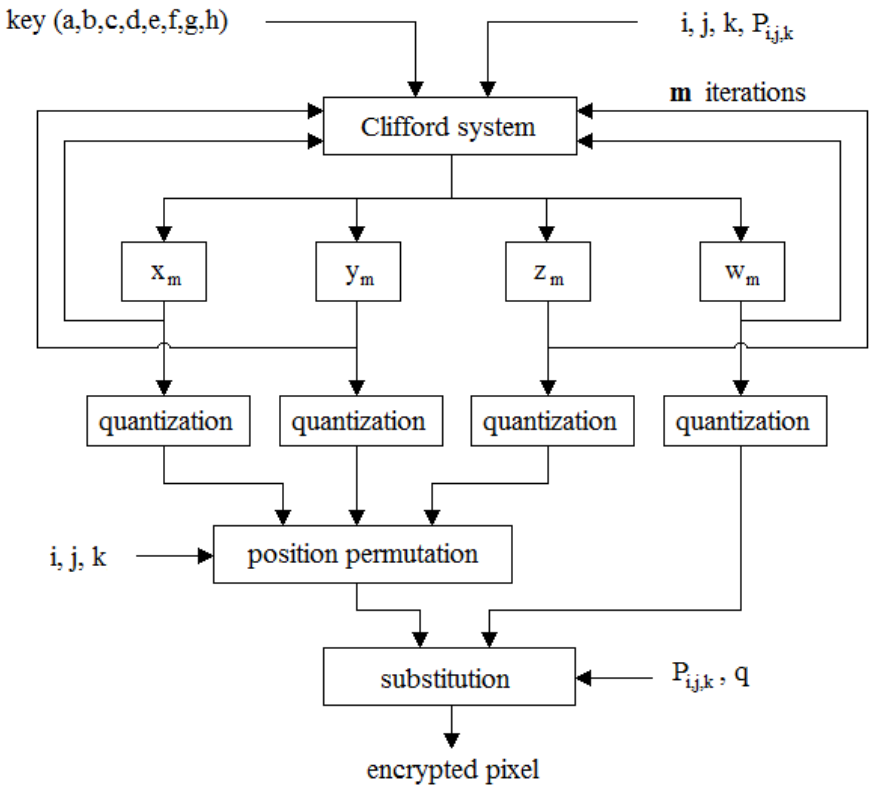
kde U je velké číslo, které zajistí zesílení vstupu a T je hraniční hodnota.

Kvantované výstupy x_m, y_m a z_m reprezentují souřadnice pixelu p_{x_m,y_m,z_m} a w_m reprezentuje modifikační hodnotu. Tato hodnota je použita k úpravě hodnoty nově nalezeného pixelu p_{x_m,y_m,z_m} podle (6.5) a zároveň je pixel $p_{i,j,k}$ prohozen s nově vypočítanou hodnotou.

$$p_{i,j,k} \leftrightarrow p_{x_m,y_m,z_m} \oplus w_m \oplus q \tag{6.5}$$

Proměnná q je hodnota pixelu, který byl zpracován v předchozím kroku. Po každém provedení operace (6.5) je tedy nastaven jako $q = p_{i,j,k}$.

Tento proces musí být proveden pro každou hodnotu v matici P (tedy pro každý pixel v obraze) a může být proveden několikrát. Obecně platí, že více šifrovacích rund zvyšuje zabezpečení zašifrovaného obrazu. Obrázek 6.2 ukazuje zjednodušený nákres šifrovacího procesu.



Obrázek 6.2: Nákres šifrovacího procesu

Řídicí parametry Cliffordova systému zde tedy hrají roli reálných šifrovacích klíčů a souřadnice pixelů a jejich hodnoty se berou jako počáteční podmínky systému (6.2). Po několika iteracích systému jsou k dispozici hodnoty, které se použijí k šifrovacím účelům, tedy k permutaci pozice pixelu a substituci jeho hodnoty. První tři předpisy systému (6.2) se starají o permutace pozic nejen v rámci dvou-rozměrné plochy, ale i mezi barevnými rovinami obrazu. Čtvrtý předpis systému (6.2) generuje modifikační hodnoty, které upravují jasy šifrovaných pixelů.

6.2. Experimentální výsledky

Každý kryptografický systém by měl splňovat některé bezpečnostní doporučení uvedené v [26, 27]. Tato část práce analyzuje navržený kryptografický systém z hlediska zabezpečení zašifrovaných obrazů.

6.2.1. Distribuce pixelů

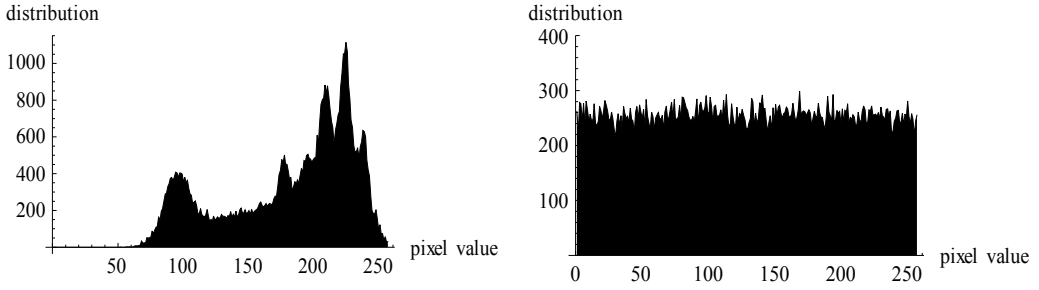
Účinnost substituční funkce lze nejefektivněji vyjádřit výslednou změnou distribuce pixelů. Použití substituce je nutné, protože samotná permutace pozice není z bezpečnostního hlediska dostačující. Kryptografický systém založený pouze na permutačních operacích může být jednoduše odhalen např. systémem ergodických fuzzy matic [28].

Tato kapitola ukazuje změny distribucí u obrazu „Lena“ o velikosti 256x256 pixelů. Obrázek 6.3 ukazuje obraz před a po zašifrování.

(a) (b)

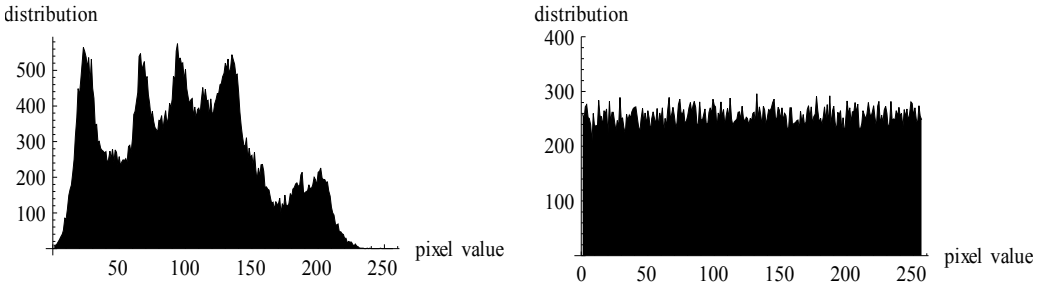
Obrázek 6.3: (a) původní obraz, (b) zašifrovaný obraz

Lze vidět, že zašifrovaný obraz je velmi zašuměný a nelze z něj vyčíst žádná konkrétní informace. Obrázky 6.4-6.6 ukazují, jak se distribuce jednotlivých barevných složek původního obrazu po zašifrování změnily. Byly zkoumány hodnoty všech barevných složek modelu RGB.



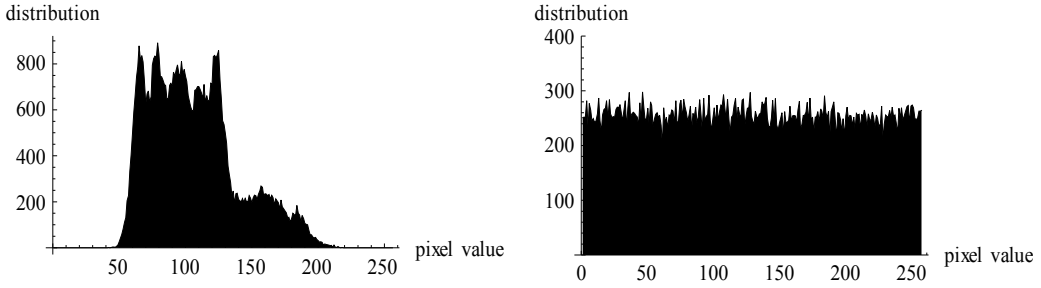
(a) (b)

Obrázek 6.4: (a) distribuce původní R složky, (b) distribuce zašifrované R složky



(a) (b)

Obrázek 6.5: (a) distribuce původní G složky, (b) distribuce zašifrované G složky



(a) (b)

Obrázek 6.6: (a) distribuce původní B složky, (b) distribuce zašifrované B složky

Je evidentní, že distribuce barevných složek zašifrovaného obrazu jsou rozdílné od původního obrazu a velmi blízké rovnoměrnému rozdělení. Rovnoměrnost zde znamená, že neexistuje žádná statistická podobnost mezi původním a zašifrovaným obrazem. Tento aspekt činí kryptografický systém rezistentní proti útoku typu „known-plaintext“.

6.2.2. Entropie obrazů

Nečitelnost a nepředvídatelnost jsou jedním z hlavních cílů kryptografického systému. Tato nepředvídatelnost může být reflektována jednou z nejčastěji používanou informační mírou – entropií.

Entropie H zdroje zpráv S může být vyjádřena jako (6.6)

$$H(S) = \sum_{i=1}^{N} P(s_i) \cdot \log \frac{1}{P(s_i)} \tag{6.6}$$

kde $P(s_i)$ reprezentuje pravděpodobnost symbolu s_i, tedy četnost výskytu hodnoty pixelu, a $\log$ je binárním logaritmem o základu *2*.

Entropie obrazu je maximální, pokud všechny hodnoty pixelů jsou distribuovány rovnoměrně. Požadovaný efekt šifrovacího procesu je dosáhnout maximální hodnoty entropie. Tabulka 6.1 zobrazuje hodnoty entropie původních obrazů a jejich zašifrovaných forem. Tyto hodnoty jsou velmi blízké maximální hodnotě entropie.

Tabulka 6.1: Hodnota entropie pro různé zdroje zpráv

Obraz	**Entropie původního obrazu**	**Entropie zašifrovaného obrazu**
Šedá škála	8	7.99737
Černá barva	0	7.99714
Lena R	7.25086	7.99742
Lena G	7.59057	7.99730
Lena B	6.92842	7.99689

Dokonce i obraz obsahující pouze pixely černé barvy, který má nulovou entropii, dosahoval po šifrovacím procesu k maximální hodnotě entropie. Znamená to tedy, že během šifrovacího procesu dochází k vysoké konfúzi a difúzi.

6.2.3. Křížová korelace obrazů

Křížová korelace patří mezi standardní metody k určení míry podobnosti dvou sérií dat. Předpokládejme dva vektory dat x_i a y_i, kde $i = 1,2,...,N$ a $E(x)$, $E(y)$ jsou střední hodnoty daných vektorů spočítaných podle (6.7).

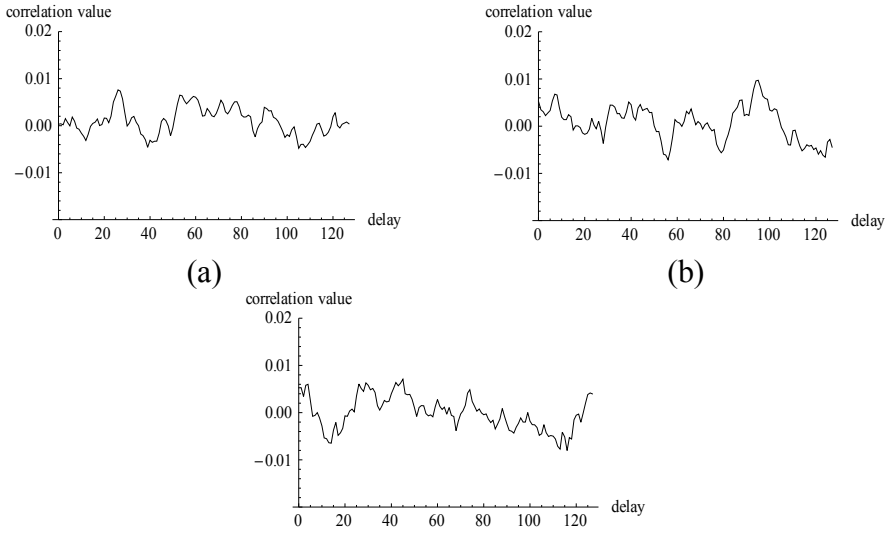
$$E(x) = \frac{1}{N} \cdot \sum_{i=1}^{N} x_i$$
$$E(y) = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i \qquad (6.7)$$

Křížová korelace r v případě posunutí d je definována jako (6.8)

$$r(d) = \frac{\sum_i (x_i - E(x)) \cdot (y_{i-d} - E(y))}{\sqrt{\sum_i (x_i - E(x))^2} \cdot \sqrt{\sum_i (y_{i-d} - E(y))^2}} \tag{6.8}$$

Jmenovatel v (6.8) normalizuje korelační koeficienty do intervalu $-1 \le r(d) \le 1$. Hraniční hodnoty intervalu značí maximální korelaci a $r(d) = 0$ značí korelaci nulovou. Záporné korelační hodnoty reprezentují korelaci v případě inverze jednoho z vektorů.

Pomocí nástroje křížové korelace lze tedy kvantitativně vyjádřit podobnost či rozdílnost dvou obrazů, které se liší posunutím. Obrázek 6.7 ukazuje korelační koeficienty křížové korelace pro posunutí $d = 0$ až $d = 128$ pro různé barevné složky původního obrazu „Lena“ a jeho zašifrované formy.



Obrázek 6.7: Křížová korelace (a) R, (b) G, (c) B složky původního a zašifrovaného obrazu

Lze vidět, že korelační koeficienty nepřesáhnou hodnotu 0.01, což značí velmi nízkou korelaci a velmi nízkou podobnost původního a zašifrovaného obrazu.

6.2.4. Křížová korelace sousedních pixelů

Sousední pixely v reálných obrazech bývají obecně velmi silně korelovány. Jedním ze základních požadavků efektivního kryptografického systému je vytvořit zašifrovaný obraz, kde sousední pixely mají velmi nízkou korelaci. Tato podkapitola analyzuje korelaci sousedních pixelů ve třech směrech – horizontálním, vertikálním a diagonálním. Z původního obrazu byla vybrána každá dvojice pixelů sousedící v příslušném směru a mezi těmito pixely byl spočítán korelační koeficient podle (6.8). Obdobný postup byl proveden i v případě zašifrovaného obrazu. Střední hodnoty korelačních koeficientů pro různé barevné složky jsou zobrazeny v Tabulkách 6.2-6.4.

Tabulka 6.2: Křížová korelace sousedních pixelů pro R složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.954094	0.002585
Vertikální	0.976929	-0.002036
Diagonální	0.929461	0.000506

Tabulka 6.3: Křížová korelace sousedních pixelů pro G složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.938597	-0.003592
Vertikální	0.968472	-0.001187
Diagonální	0.913181	0.006435

Tabulka 6.4: Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.922301	0.000719
Vertikální	0.951445	0.000225
Diagonální	0.892751	-0.004336

Navržený kryptografický systém tedy velmi efektivně dekoreluje sousední pixely původního obrazu ve všech barevných složkách. Pro srovnání s ostatními kryptografickými systémy založenými na deterministickém chaosu je uvedena Tabulka 6.5, která uvádí korelační koeficienty zašifrovaného obrazu „Lena“ pro R složku.

Tabulka 6.5: Srovnání křížové korelace s jinými kryptografickými systémy

Směr	**Navržený systém**	**Systém [9]**	**Systém [23]**
Horizontálně	0.002585	0.005776	0.002637
Vertikálně	-0.002036	0.028434	0.009177
Diagonálně	0.000506	0.020662	0.003429
Směr	**Systém [25]**	**Systém [29]**	**Systém [31]**
Horizontálně	0.030800	-0.014200	0.01589
Vertikálně	0.030400	-0.007400	0.06538
Diagonálně	0.031700	-0.018300	0.03231
Směr	**Systém [32]**	**Systém [33]**	
Horizontálně	0.01183	0.00261	
Vertikálně	0.00016	0.00371	
Diagonálně	0.01480	0.00403	

Lze vidět, že v případě použití navrženého kryptografického systému získáme zašifrovaný obraz, kde sousední pixely jsou nejméně korelovány. Navržený systém má tedy velmi efektivní difúzní a konfúzní vlastnosti.

6.2.5. Citlivost klíčů

Protože za klíče považujeme řídicí parametry Cliffordova systému, pak minimální změna v klíči by měla zajistit rozdílný výstup kryptografického systému. Pro experimenty byly vytvořeny dvě sady klíčů (parametrů) chaotického systému (6.2):

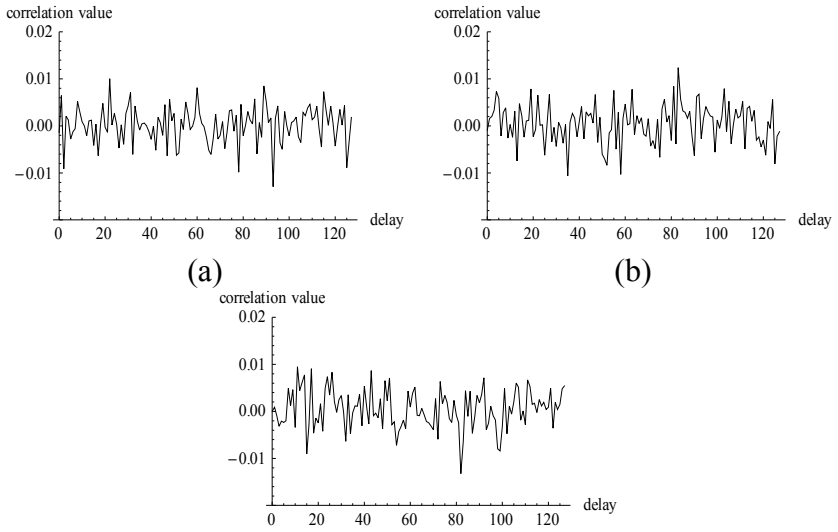
$$a = -1.4,\ b = 1.6,\ c = -1.8,\ d = 0.9,\quad e = 1.3,\ f = 1.7,\ g = 0.4,\ h = 1.2 \tag{6.9}$$

$$a = -1.4,\ b = 1.60000001,\ c = -1.8,\ d = 0.9,\quad e = 1.3,\ f = 1.7,\ g = 0.4,\ h = 1.2 \tag{6.10}$$

Sady klíčů (6.9) a (6.10) jsou velmi podobné, jediný parametr je odlišný a to s velmi malou odchylkou.

První test citlivosti klíčů

První test citlivosti klíčů spočívá v zašifrování obrazu „Lena“ pomocí klíčů (6.9) a zašifrování obrazu „Lena“ pomocí klíčů (6.10). Následně se spočítá křížová korelace těchto zašifrovaných obrazů. Obrázek 6.8 ukazuje právě tuto křížovou korelaci pro jednotlivé barevné složky.



Obrázek 6.8: Křížová korelace (a) R, (b) G, (c) B složky dvou zašifrovaných obrazů

Koeficienty křížové korelace obou obrazů jsou velmi nízké. To značí velmi malou podobnost mezi těmito obrazy. Procentuální rozdílnost barevných složek při použití různých šifrovacích klíčů je vyčíslena v Tabulce 6.6.

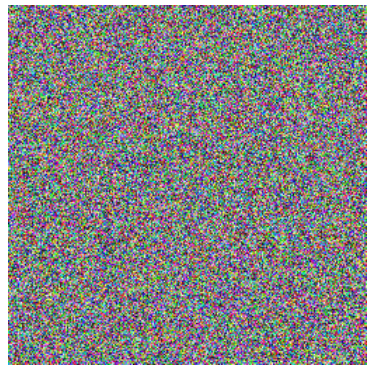
Tabulka 6.6: Rozdílnost zašifrovaných obrazů pro různé šifrovací klíče

RGB	Rozdílnost zašifrovaných obrazů
R	99.591064%
G	99.617004%
B	99.624633%

Různé šifrovací klíče tedy zajistí generaci naprosto rozdílných zašifrovaných obrazů.

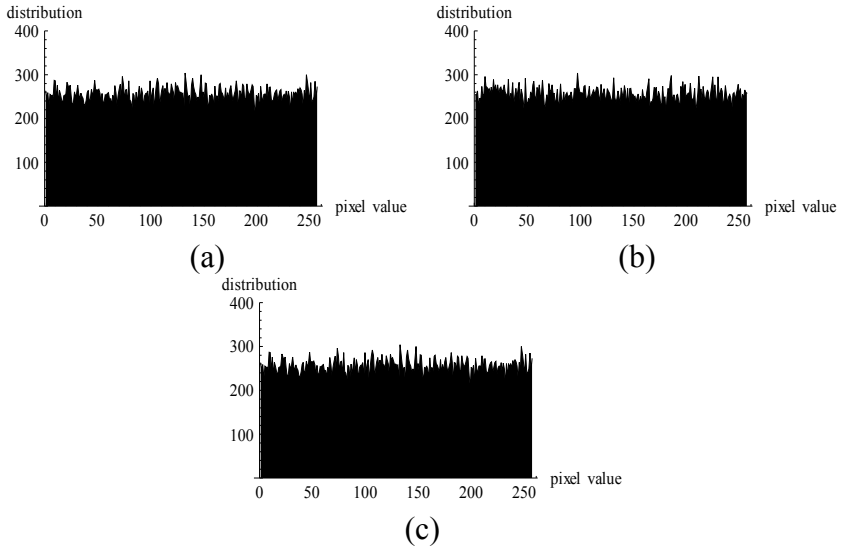
Druhý test citlivosti klíčů

Druhý test citlivosti klíčů spočívá v zašifrování obrazu „Lena“ na základě klíčů (6.9) a následném dešifrování klíči (6.10). Dešifrování tedy probíhá klíči, kde jeden parametr je od toho správného rozdílný. Obrázek 6.9. ukazuje, že nesprávně dešifrovaný obraz je nečitelný a stále odpovídá zašuměnému obrazu.



Obrázek 6.9: Nesprávně dešifrovaný obraz

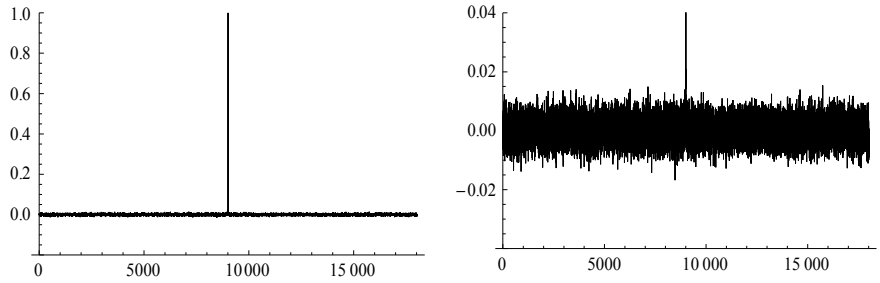
Obrázek 6.10 ukazuje distribuci různých barevných složek nesprávně dešifrovaného obrazu.



Obrázek 6.10: Distribuce (a) R, (b) G, (c) B složky nesprávně dešifrovaného obrazu

Lze vidět, že distribuce pixelů nesprávně dešifrovaného obrazu je opět velmi blízká rovnoměrnosti. Pokud se tedy v klíčích pro dešifrování nalézá velmi malá odchylka, nelze obraz rekonstruovat ani do jeho přibližné původní formy. Toto je velmi dobrá vlastnost proti útoku hrubou silou, protože ani ve velmi blízkém okolí správného klíče není nijak patrné, že se útočník ke správnému klíči blíží. Pokud se útočník netrefí do naprosto přesné hodnoty klíče, pak není schopen ani říci, ve které oblasti se pravděpodobně správný klíč nalézá. Toto tvrzení je podloženo následujícím experimentem.

Vezměme parametr b ze sady dešifrovacích klíčů (6.10). Poté se spočítají koeficienty křížové korelace mezi původním obrazem a dešifrovaným obrazem pro $b = 1.59999999999$ až $b = 1.60000000001$ po kroku $1\mathrm{E}\text{-}15$. Křížová korelace bude maximální při správné hodnotě parametru $b = 1.6$. Obrázek 6.11 ukazuje korelaci pro různé hodnoty parametru b.



(a) (b)

Obrázek 6.11: (a) křížová korelace pro různé klíče, (b) detailní pohled

Hrot uprostřed grafu značí maximální korelaci, všechny ostatní obrazy dešifrované nesprávnými klíči (tedy nesprávným parametrem b) jsou k původnímu obrazu velmi málo korelované. Z grafu není patrný žádný trend zvyšování korelace ani v případě, že se parametr blíží k jeho správné hodnotě.

6.2.6. Citlivost obrazu

V diferenciálních analýzách používá kryptoanalytik zprávy, které se mezi sebou v některých hodnotách liší. Má-li k dispozici odpovídající zašifrované zprávy, může analyzovat rozdíly mezi těmito zprávami a nalézt vztahy mezi původní zprávou a její zašifrovanou formou. Proto pro větší zabezpečení zprávy by měly být rozdíly podobných původních obrazů po zašifrování velmi velké. Citlivost obrazu je nejčastěji reflektována dvěma ukazateli: NPCR a UACI.

Míra počtu změněných pixelů (NPCR) je ukazatel, který vyjadřuje počet rozdílných pixelů v zašifrovaných formách, když je změněna například hodnota pouze jednoho pixelu v původních obrazech. Pokud je NPCR větší, pak obraz má vyšší citlivost a systém je bezpečnější proti diferenciální analýze.

Mějme dva zašifrované obrazy A a B stejné velikosti. NPCR barevné složky k je poté dána podle (6.11)

$$NPCR_k = \frac{\sum_{i,j} F(i,j,k)}{W \times H} \tag{6.11}$$

kde W a H jsou výška a šířka obrazů a $F(i,j,k)$ je bipolární pole definované jako (6.12)

$$F(i,j,k)=\begin{cases}0, & A(i,j,k)=B(i,j,k)\\ 1, & A(i,j,k)\neq B(i,j,k)\end{cases} \tag{6.12}$$

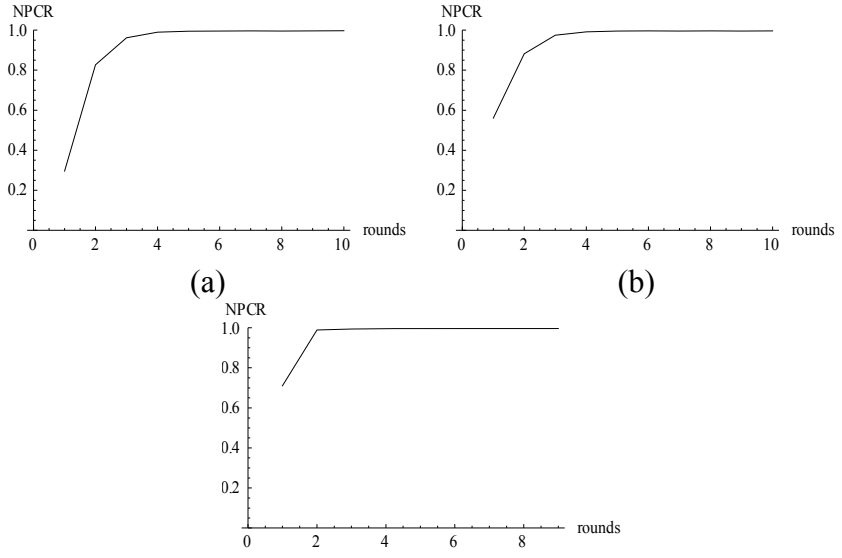
Výrazy $A(i,j,k)$ a $B(i,j,k)$ reprezentují hodnotu pixelu příslušného obrazu na souřadnici (i,j,k).

Druhým ukazatelem citlivosti obrazu je unifikovaná průměrná změna intenzity (UACI). Čím vyšší je hodnota ukazatele UACI, tím více je kryptografický systém rezistentní vůči diferenciální analýze.

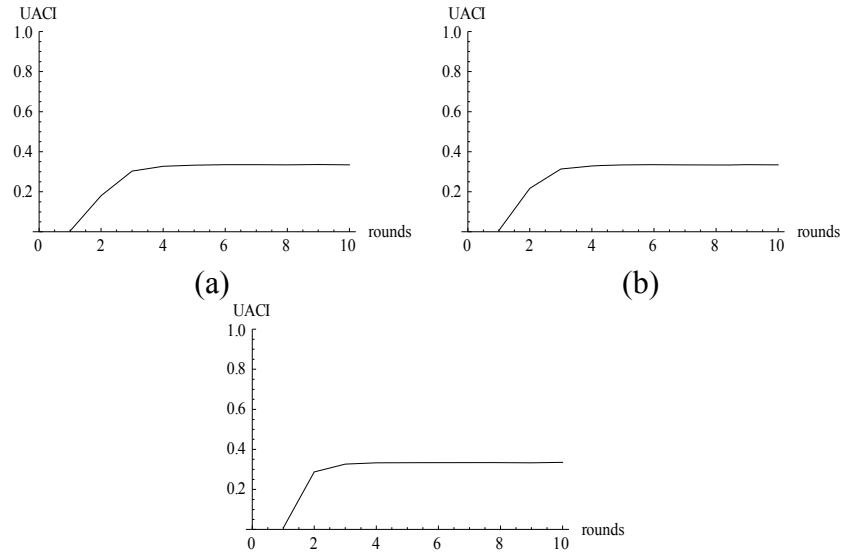
UACI pro barevnou složku k je definován jako (6.13)

$$UACI_k=\frac{1}{W\times H}\left[\sum_{i,j}\frac{\left|A(i,j,k)-B(i,j,k)\right|}{255}\right] \tag{6.13}$$

Pro následující test předpokládejme dva velmi podobné obrazy. Tyto obrazy se liší pouze v pixelu na souřadnici (2,3,0). V hodnotě tohoto pixelu je u druhého obrazu invertován nejméně signifikantní bit. Následně jsou oba obrazy zašifrovány a spočítány ukazatele NPCR a UACI. Obrázek 6.11 a Obrázek 6.12 zobrazuje vývoj NPCR a UACI v závislosti na šifrovacích rundách.



Obrázek 6.12: NPCR pro (a) R, (b) G, (c) B složku



Obrázek 6.13: UACI pro (a) R, (b) G, (c) B složku

Lze vidět, že po pěti šifrovacích rundách dosáhl kryptografický systém dostatečného zabezpečení šifrovaného obrazu proti diferenciální analýze. Nejpomalejší vzestup má R složka, to je dáno ale především tím, že komponenty R složky se šifrují jako první. Před samotným šifrováním B složky již v ní existuje určitá změna díky prohazování pixelů mezi barevnými rovinami při šifrování předešlých barevných složek. Tabulky 6.7-6.9. vyjadřují ukazatele NPCR a UACI numericky vzhledem k šifrovacím rundám pro určité barevné složky.

Tabulka 6.7: NPCR a UACI pro R složku

	Šifrovací rundy				
	1	**2**	**3**	**4**	**5**
NPCR	0.295944	0.826721	0.961456	0.989685	0.994415
UACI	0.004323	0.179768	0.302886	0.326836	0.332527

Tabulka 6.8: NPCR a UACI pro G složku

	Šifrovací rundy				
	1	**2**	**3**	**4**	**5**
NPCR	0.560714	0.882156	0.974090	0.991287	0.995178
UACI	0.006608	0.216789	0.313346	0.328897	0.334682

Tabulka 6.9: NPCR a UACI pro B složku

	Šifrovací rundy				
	1	**2**	**3**	**4**	**5**
NPCR	0.745025	0.953582	0.988601	0.993927	0.995864
UACI	0.008759	0.2871348	0.326093	0.333011	0.333932

Následující Tabulka 6.10 a Tabulka 6.11 srovnávají navržený kryptografický systém se systémy jinými z hlediska NPCR a UACI. Lze vidět, že systémy [23] a [34] dosáhnou vysokých hodnot velmi rychle. Tyto systémy byly navrženy tak, aby dosáhly co největší difúze v co nejkratším čase. Většina ostatních systémů nemá ukazatele NPCR a UACI zveřejněny v číselné formě, proto je nelze s navrženou šifrou srovnávat.

Tabulka 6.10: Srovnání NPCR s jinými systémy

	Šifrovací rundy				
	1	**2**	**3**	**4**	**5**
Navržený	0.295944	0.826721	0.961456	0.989685	0.994415
[22]	0.000179	0.011070	0.438816	0.980228	0.995907
[23]	0.668423	0.996117	0.995956	0.996117	0.996113
[34]	0.995300	0.996100	0.995600	0.996700	0.995500

Tabulka 6.11: Srovnání UACI s jinými systémy

	Šifrovací rundy				
	1	**2**	**3**	**4**	**5**
Navržený	0.004323	0.179768	0.302886	0.326836	0.332527
[22]	0.000004	0.002750	0.119449	0.298463	0.331217
[23]	0.202745	0.334731	0.334844	0.335290	0.335271
[34]	0.196100	0.333900	0.334100	0.335500	0.335800

6.2.7. Prostor klíčů

Spolehlivý kryptografický systém musí být rezistentní proti útokům hrubou silou, kdy je prohledáván celý prostor klíčů a hledán správný klíč pro dešifrování. Tento prostor by tedy měl být co nejširší, aby zde existovalo dostatečné množství klíčů, které musí kryptoanalytik vyzkoušet. Přesnost klasického procesoru je 16 číslic v desítkové soustavě. Počet různých kombinací jednoho řídicího parametru (klíče) je tedy 10^{16}. To přibližně odpovídá velikosti prostoru klíčů 2^{53}. Navržená šifra používá osm různých řídicích parametrů, proto je prostor klíčů velký 2^{424}. Za šifrovací klíče lze případně považovat i počet iterací Cliffordova systému a počet šifrovacích rund. Prostor klíčů je tedy dostatečně široký na to, aby zabránil útokům hrubou silou. Tabulka 6.12 obsahuje srovnání prostoru klíčů navrženého kryptografického systému s jinými systémy. Lze vidět, že navržený systém má nejširší prostor klíčů.

Tabulka 6.12: Srovnání velikosti prostoru klíčů s jinými systémy

Kryptografický systém	Prostor klíčů
Navržený	2^{424}
[9]	2^{128}
[25]	2^{256}
[29]	2^{232}
[30]	2^{158}
[34]	2^{141}

6.2.8. Výkonnost šifrovacího algoritmu

Navržený kryptografický systém má velmi dobré difúzní a konfúzní vlastnosti, nicméně dosažení těchto silných stránek si vyžaduje velkou výpočetní náročnost. Tato náročnost je nejslabší stránkou celého kryptografického systému, protože nelze navržený algoritmus využít v aplikacích pracujících v reálném čase. Výkonnost algoritmu z hlediska výpočetní náročnosti je nejvíce ovlivněna iteracemi Cliffordova systému a počtem šifrovacích rund. Ze zkoumání citlivosti obrazu bylo zjištěno, že zabezpečení zašifrovaného obrazu je přímo úměrné počtu šifrovacích rund.

V následujícím testu byl šifrován 24-bitový obraz „Lena“ o velikosti 256x256 pixelů na procesoru AMD Turion 64 X2 s taktovací frekvencí 2.0GHz a operační pamětí 2.0GB RAM. Tabulka 6.13 obsahuje dobu trvání šifrovacího procesu v sekundách pro různé iterace Cliffordova systému a pro různý počet šifrovacích rund.

Tabulka 6.13: Výkonnost šifrovacího algoritmu

Iterace / Rundy	**1**	**5**	**10**	**15**	**20**	**25**
1	0.173	0.590	1.125	1.552	2.938	2.526
2	0.382	1.181	2.159	3.872	4.133	5.111
3	0.569	1.807	3.208	4.621	6.104	7.643
4	0.763	2.297	4.179	6.105	8.132	10.501
5	0.858	2.290	5.328	7.651	10.113	12.567

Testy prokázaly, že nejvyšší rychlost zpracování dat, kterou lze dosáhnout na výše zmíněném procesoru, je 1.08 MB/s. Pro dostatečné zabezpečení je ovšem nutné nastavit iterace Cliffordova systému a počet šifrovacích rund na 5. Rychlost zpracování dat se pak sníží na 83.8 kB/s. Navržený kryptografický systém je tedy vhodný spíše pro archivační účely.

7. OPTIMALIZACE RYCHLOSTI KRYPTOGRAFICKÉHO SYSTÉMU

Z Tabulky 6.13 lze usoudit, že navržený šifrovací algoritmus není vhodný pro aplikace, které vyžadují šifrování obrazu v reálném čase. Obraz a obecně většina multimediálních signálů obsahují velké množství dat. Protože souřadnice pixelů a jejich hodnoty jsou považovány za počáteční podmínky Cliffordova systému, šifrovací proces musí proběhnout pro každý pixel. Z důvodu velkého množství dat v obrazu a nutnosti postupného šifrování všech těchto dat je tedy algoritmus výpočetně velmi náročný. Tato kapitola nastiňuje způsob pro snížení výpočetní náročnosti a možnosti zavedení ztrátové komprese.

7.1. Waveletová transformace

Pro reprezentaci nestacionárních signálů je často velmi vhodné použít transformaci, která signál převádí z časové do časově-frekvenční oblasti. Tímto způsobem lze získat příznaky, pomocí kterých lze signál lépe klasifikovat. Časově-frekvenční analýza vypočte nejen frekvenční složky, ale také lokalizuje dobu jejich výskytu. Mezi nejznámější lineární časově-frekvenční analýzy patří krátkodobá Fourierova transformace a waveletová transformace.

Waveletová transformace je definována jako (7.1)

$$W(s,\tau) = \int f(t) \cdot \psi^*_{s,\tau}(t) dt \tag{7.1}$$

kde $*$ je označením komplexně sdružené proměnné. Tento vztah reprezentuje rozložení funkce $f(t)$ do sady bázových funkcí $\psi_{s,\tau}(t)$, tzv. waveletů [35].

Wavelety jsou generovány z jediné bázové funkce $\psi(t)$, která se nazývá mateřský wavelet. Tuto bázovou funkci lze zapsat podle (7.2)

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \cdot \psi\left(\frac{t-\tau}{s}\right) \tag{7.2}$$

Pomocí měřítka s je možné měnit šířku waveletu. Normalizace $\sqrt{s}$ zajišťuje, že bude mít wavelet pro všechna měřítka normalizovanou energii. Parametrem τ se mění poloha waveletu na časové ose [35].

Z rovnice (7.1) je patrné, že výpočet spojité waveletové transformace je velmi redundantní. Spojitá změna měřítka s a spojitá změna polohy τ povede k výpočtu nekonečného množství waveletových koeficientů. V praxi se proto často používá diskrétní transformace, která je speciálně vzorkovaná a jejíž předpis je dán jako (7.3)

$$\psi_{p,k}(t) = \frac{1}{\sqrt{|2^p|}} \cdot \psi\left(\frac{t - 2^p \cdot k}{2^p}\right) \tag{7.3}$$

kde parametry p a k odpovídají měřítku a posunutí (7.4)

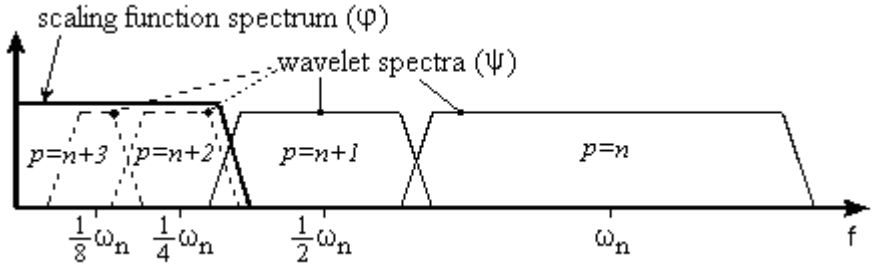
$$\begin{aligned} s &= 2^p \\ \tau &= 2^p \cdot k \end{aligned} \tag{7.4}$$

Díky takto zavedené ortonormalitě umožňuje waveletová transformace neredundantní dekompozici signálu, tzv. analýzu s mnoha rozlišeními.

Diskrétní waveletová funkce ψ se chová jako horní pásmová propust, která filtruje signál. V dalším měřítku je vždy filtrována pouze horní polovina pásma předchozí dolnofrekvenční části signálu. Pokud ovšem zkrátíme wavelet v čase o polovinu, pokryjeme pouze polovinu zbývajícího spektra. Z tohoto důvodu by bylo potřeba opět nekonečného počtu waveletů. Řešením je zavedení dolní pásmové propusti. Tato propust se nazývá měřítková funkce a je definována jako (7.5) [36]

$$\varphi_{p,k}(t) = \sum_{p,k} W(p,k) \cdot \psi_{p,k}(t) \tag{7.5}$$

Lze vidět, že měřítková funkce pokrývá spektrum všech waveletů až do hodnoty měřítka p, zbytek je vyplněn samotnými wavelety ψ. Celou problematiku půlení spektra a využití měřítkové funkce názorně ukazuje Obrázek 7.1.



Obrázek 7.1: Využití měřítkové funkce (převzato z [36])

Pokud tedy vlnku ψ chápeme jako horní pásmovou propust a měřítkovou funkci φ jako dolní propust, pak řadu dilatovaných vlnek můžeme společně s měřítkovou funkcí považovat za iterační banku FIR filtrů, tedy filtrů s konečnou impulsní odezvou, a samotnou vlnkovou transformaci za průchod signálu touto bankou. Výhodou této metody je, že je zapotřebí pouze dvou filtrů - filtru typu horní propust a dolní propust. Nevýhodou je fixní pokrytí signálního spektra. Tento druh analýzy se nazývá rychlá waveletová transformace [36].

Kvadraturně zrcadlové filtry H a G se čtyřmi koeficienty realizující dolní propust a horní propust mohou být definovány jako matice (7.6) a (7.7)

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & \cdot & \cdot & \cdot & \cdot & & \\ & & & & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & & & & & h_0 & h_1 \end{pmatrix} \tag{7.6}$$

$$G = \begin{pmatrix} g_2 & g_3 & & & & & g_0 & g_1 \\ g_0 & g_1 & g_2 & g_3 & & & & \\ & & \cdot & \cdot & \cdot & \cdot & & \\ & & & & g_0 & g_1 & g_2 & g_3 \end{pmatrix} \tag{7.7}$$

kde koeficienty filtru G jsou vytvořeny podle (7.8).

$$g_n = (-1)^n \cdot h_{N-1-n} \tag{7.8}$$

Waveletová transformace spočívá v aplikaci vektoru X, který obsahuje komponenty signálu, na matice H a G podle (7.9). Výsledkem je vektor aproximace A a vektor waveletových koeficientů C.

$$X \cdot \begin{pmatrix} H \\ G \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \tag{7.9}$$

Vektor A obsahuje složky nižších frekvencí a vektor C složky vyšších frekvencí zpracovaného signálu.

Zpětnou waveletovou transformaci lze formulovat následovně (7.10)

$$\begin{pmatrix} A \\ C \end{pmatrix} \cdot \begin{pmatrix} H_R \\ G_R \end{pmatrix} = X \tag{7.10}$$

kde H_R a G_R jsou filtry pro zpětnou transformaci. Aby byla rovnice (7.10) splněna, musí platit (7.11)

$$\begin{pmatrix} H \\ G \end{pmatrix} \cdot \begin{pmatrix} H_R \\ G_R \end{pmatrix} = I \tag{7.11}$$

Výsledkem součinu matic tedy musí být jednotková matice, což se zajistí velmi jednoduše, pokud se využije vlastnosti ortogonality (7.12)

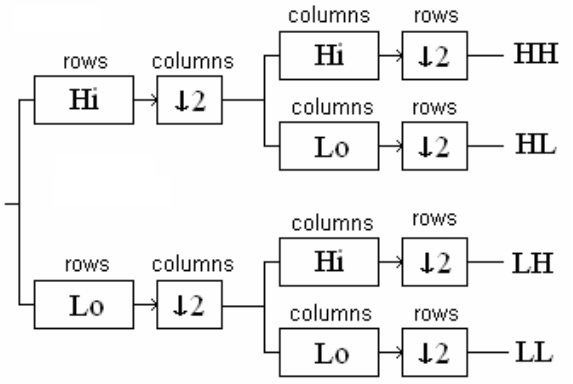
$$\begin{pmatrix} H \\ G \end{pmatrix} \cdot \begin{pmatrix} H \\ G \end{pmatrix}^T = I \tag{7.12}$$

kde $\begin{pmatrix} H \\ G \end{pmatrix}^T$ je transponovaná matice $\begin{pmatrix} H \\ G \end{pmatrix}$.

7.2. Dyadická dekompozice obrazu

Dyadická dekompozice je velmi často používána při zpracování multimediálních signálů. Signál se nechá projít bankou filtrů typu horní propust k analýze vyšších frekvencí a bankou filtrů typu dolní propust k analýze nižších frekvencí. Signál se tedy dělí na aproximaci (reprezentován nižšími frekvencemi zpracovaného signálu) a detailnější informaci (reprezentován vyššími frekvencemi zpracovaného signálu). Potom následuje podvzorkování, pomocí kterého se odstraní část vzorků v signálu.

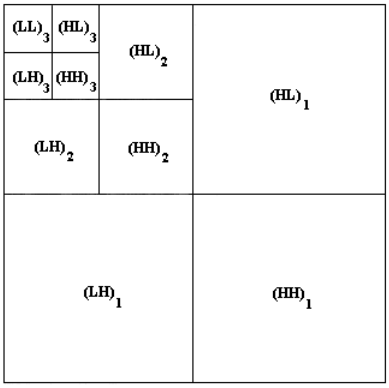
Základní strukturu dyadické dekompozice lze vidět na Obrázku 7.2. Bloky *Hi* a *Lo* reprezentují impulsní odezvu hornopropustního filtru a dolnopropustního filtru a *2*↓ znamená podvzorkování.



Obrázek 7.2: Základní struktura dyadické dekompozice obrazu

Nejprve se tedy transformují všechny řádky obrazu. Takto zanalyzovaná data se sloupcově podvzorkují a následně se transformují sloupce těchto dat. Po řádkovém podvzorkování je tedy provedena dyadická dekompozice první úrovně a ve výsledku jsou k dispozici 4 sady koeficientů: LL - aproximace obrazu, LH - detaily obrazu v horizontálním směru, HL - detaily obrazu ve vertikálním směru, HH - detaily obrazu v diagonálním směru.

Podle celkové hloubky dekompozice a podle použitého druhu transformace se dosahuje různé přesnosti popisu obrazu resp. popisu rozložení energie v obrazu. Pro mnoho-úrovňovou dekompozici signálu se často používá nestandardní dyadická dekompozice, u které se vždy rekurzivně analyzuje pouze aproximační část (LL pásmo) z předchozího kroku dekompozice. Na Obrázku 7.3 je schématicky zobrazena dekompozice obrazu třetí úrovně, která se často používá v oblasti zpracování statických obrazů.



Obrázek 7.3: Dyadická dekompozice třetí úrovně

Obraz může být znovu rekonstruován a transformován zpět do časové oblasti. Pokud jsou pásma LH, HL nebo HH vynulovány, obraz bude rekonstruován bez příslušných detailů. Tato vlastnost se používá při odstraňování šumu ze signálů nebo při ztrátové kompresi dat.

7.3. Úprava kryptografického systému

Do stávajících algoritmů z kapitoly *6.1.Návrh kryptografického systému* byla začleněna waveletová transformace pro extrakci signifikantních informací z obrazu. Pro rychlou transformaci byla vybrána nejjednodušší bázová funkce - Haarův wavelet zobrazený na Obrázku 7.4. Tento wavelet má nejvyšší zkreslení a tím také nejmenší účinnost při waveletové analýze. Jeho největší výhodou je ovšem nejnižší počet koeficientů. Při použití Harrova waveletu se pracuje se 2 koeficienty, proto je tento typ transformace výpočetně nejméně náročný. Koeficienty matice H jsou uvedeny v (7.13)

$$h_0 = \frac{1}{\sqrt{2}}, \quad h_1 = \frac{1}{\sqrt{2}} \tag{7.13}$$

přičemž platí (7.14), což dokazuje ortonormalitu bázové funkce.

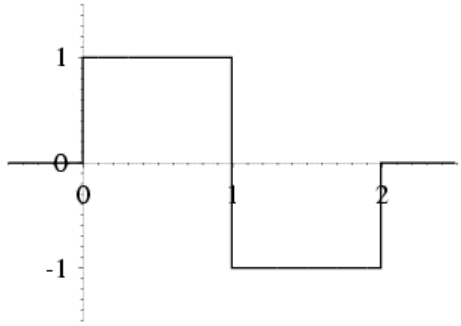
$$h_0^2 + h_1^2 = 1 \tag{7.14}$$

Koeficienty matice G jsou vytvořeny podle (7.8) a mají tedy tvar (7.15)

$$g_0 = h_0, \quad g_1 = -h_1 \tag{7.15}$$

přičemž platí, že bázová funkce má nulovou střední hodnotu (7.16).

$$g_0^2 + g_1^2 = 0 \tag{7.16}$$



Obrázek 7.4: Haarův wavelet

Iterativní banka filtrů vytvořená z Haarova waveletu je použita pro dyadickou dekompozici obrazu do nejvyšší úrovně. Uvažujme tedy opět tří-rozměrnou matici P, která reprezentuje obraz. Matice P obsahuje hodnoty pixelů $p_{i,j,k} \in P$ daného obrazu, kde $i = 0,1,2,...,W$, $j = 0,1,2,...,H$ a $k = 0,1,2,...,D$. Rozměry W, H a D určují šířku, výšku a hloubku matice P. Matice P je dyadicky dekomponována a převedena do waveletové oblasti. Výsledná tří-rozměrná matice C obsahuje waveletové koeficienty $c_{i,j,k}$, které jsou vstupními daty pro kryptografický systém.

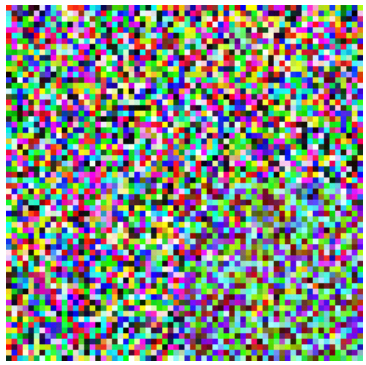
V následujících experimentech je ovšem uvažována pouze submatice $C_0 \subset C$ s koeficienty $c_{m,n,k} \in C_0$, kde $m \in (0,1,2,...,\frac{W}{4})$ a $n \in (0,1,2,...,\frac{H}{4})$ a $k = 0,1,2,...,D$. Znamená to tedy, že pouze šestnáctina všech koeficientů je šifrována. Přesto tato submatice obsahuje nejdůležitější koeficienty, které představují samotnou aproximaci a některé detaily zpracovávaného obrazu. Šifrovací proces je tedy zrychlen, pokud je zpracována pouze submatice C_0. Nicméně, použití pouhé aproximace a zanedbání detailů má velký dopad na celkovou kvalitu rekonstruovaného obrazu. Tato ztráta informace je akceptovatelná pouze v některých aplikacích, jako je například videokonference.

7.4. Experimentální výsledky

7.4.1. Změna waveletové oblasti

Tato kapitola ukazuje změny waveletové oblasti u obrazu „Lena“ o velikosti 256x256 pixelů. Obrázek 7.5 ukazuje obraz před zašifrováním, po zašifrování a následném správném dešifrování.

(a)

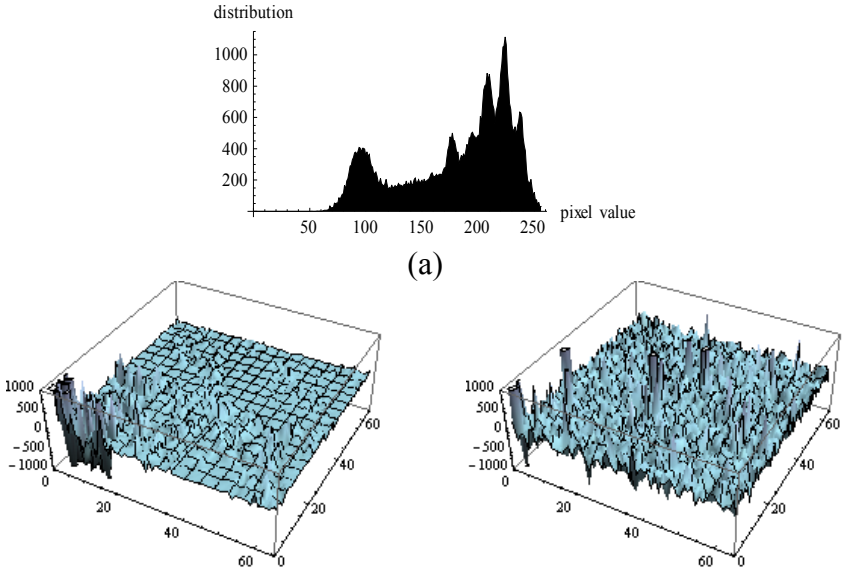


(b)

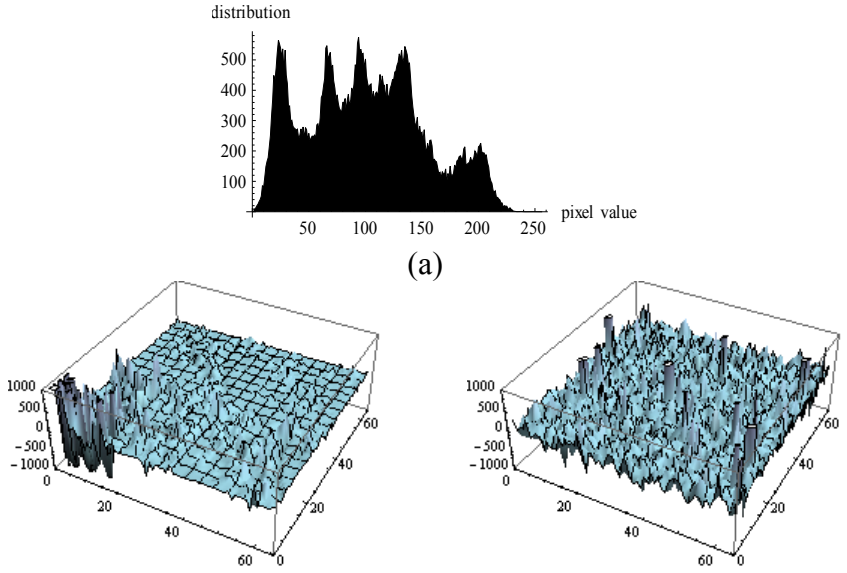
(c)

Obrázek 7.5: (a) původní obraz, (b) zašifrovaný obraz, (c) dešifrovaný obraz

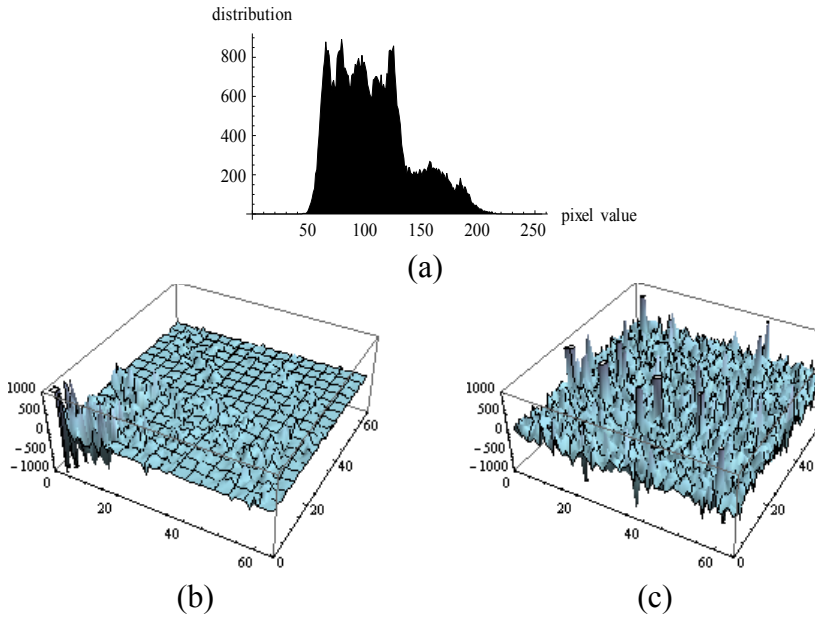
Lze vidět, že zašifrovaný obraz je opět zašuměný a dešifrovaný obraz nese známky pixelizace, která je způsobena zanedbáním velkého množství waveletových koeficientů. Obrázky 7.6-7.8 ukazují, jak se distribuce jednotlivých barevných složek původního obrazu měnily. Pro názornost jsou přidány i vzorky waveletových oblastí.



Obrázek 7.6: (a) distribuce původní R složky, (b) původní waveletová oblast R složky, (c) zašifrovaná waveletová oblast R složky



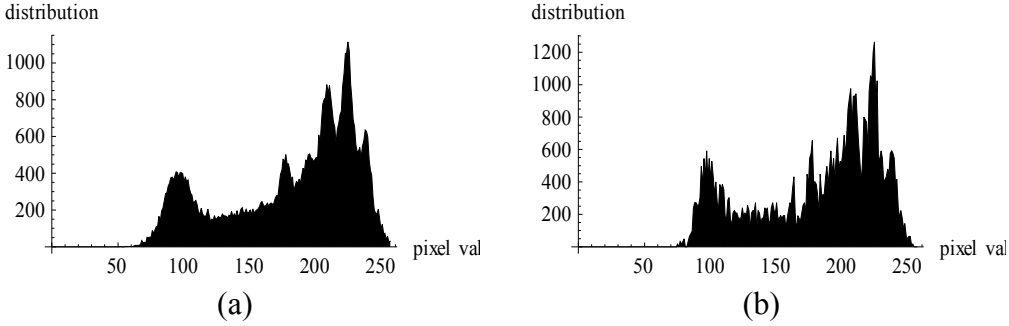
Obrázek 7.7: (a) distribuce původní G složky, (b) původní waveletová oblast G složky, (c) zašifrovaná waveletová oblast G složky



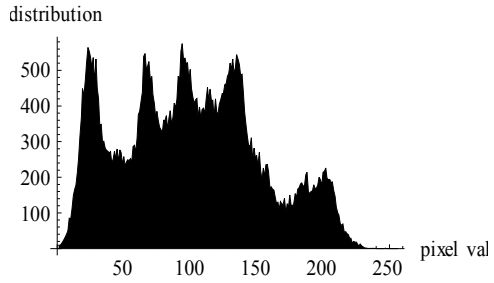
Obrázek 7.8: (a) distribuce původní B složky, (b) původní waveletová oblast B složky, (c) zašifrovaná waveletová oblast B složky

Waveletové koeficienty mají po šifrovacím procesu různé hodnoty a jsou rozmístěny po celé waveletové oblasti (submatice C_0 obsahuje různé hodnoty koeficientů na všech pozicích).

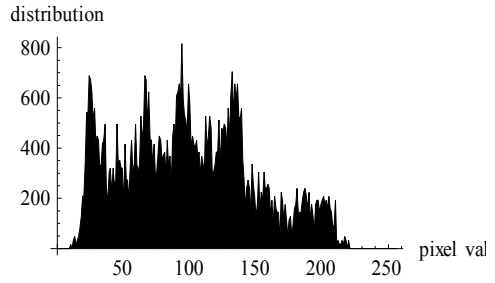
Když je obrázek dešifrován správnou sadou klíčů, získáme aproximaci původního obrazu. Ztráta informace je subjektivně evidentní z Obrázku 7.5 (c). Obrázky 7.9-7.11 srovnávají distribuci původního obrazu a obrazu dešifrovaného.



Obrázek 7.9: (a) distribuce původní R složky, (b) distribuce dešifrované R složky,

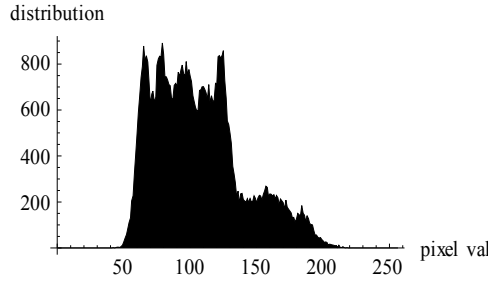


(a)

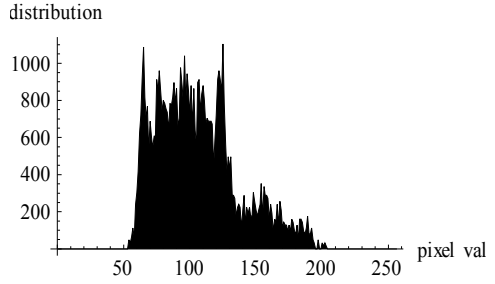


(b)

Obrázek 7.10: (a) distribuce původní G složky, (b) distribuce dešifrované G složky



(a)



(b)

Obrázek 7.11: (a) distribuce původní B složky, (b) distribuce dešifrované B složky

Lze vidět, že distribuce je mírně rozdílná. Tato ztráta informace je dána zpracováním submatice C_0, která neobsahovala detaily původního obrazu. Nicméně, všechny dešifrované waveletové koeficienty submatice C_0 jsou stejné jako před samotným šifrovacím procesem.

Ztrátu informace mezi původním a dešifrovaným obrazem můžeme objektivně reflektovat také dvěma ukazateli: střední kvadratickou odchylkou (MSE) (7.17) a špičkovým odstupem signálu od šumu (PSNR) (7.18). Tyto dva ukazatele pro původní obraz P a dešifrovaný obraz R jsou uvedeny v Tabulce 7.1.

$$MSE = \frac{1}{W \cdot H \cdot D} \sum_{i=0}^{W} \sum_{j=0}^{H} \sum_{k=0}^{D} \left\| P(i,j,k) - R(i,j,k) \right\|^2 \tag{7.17}$$

$$PSNR = 20 \cdot \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \tag{7.18}$$

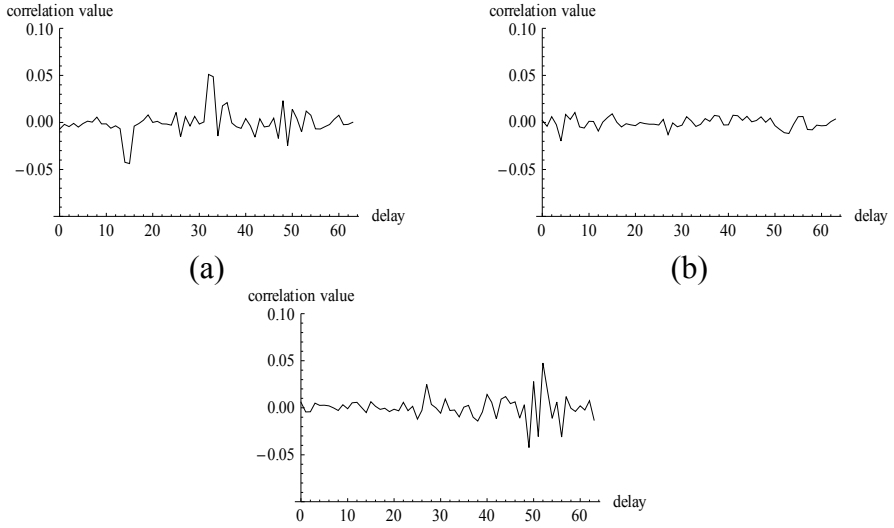
Tabulka 7.1: MSE a PSNR pro původní a dešifrovaný obraz

MSE [-]	14.7309
PSNR [dB]	24.7662

Nežádoucí jevy způsobené ztrátou informace (pixelizace a velká distorze) lze potlačit volbou vhodnější waveletové funkce, která zajistí hladkou rekonstrukci obrazu. Pro aplikace zpracování obrazu se velmi často používají wavelety Daubechies a Coiflet.

7.4.2. Testy křížové korelace

Pomocí křížové korelace byla také zkoumána podobnost původní a zašifrované waveletové oblasti. Tyto korelace jsou pro různé barevné složky zobrazeny na Obrázku 7.12. Testy ukázaly nízkou korelaci waveletových koeficientů.



Obrázek 7.12: Křížová korelace (a) R, (b) G, (c) B složky původní a zašifrované waveletové oblasti

Následující Tabulky 7.2-7.4 obsahují korelační koeficienty sousedních waveletových koeficientů. Z výsledků lze vidět, že kryptografický systém dokáže waveletové koeficienty v jejich oblasti efektivně dekorelovat.

Tabulka 7.2: Křížová korelace sousedních koeficientů pro R složku

Směr	Původní oblast	Zašifrovaná oblast
Horizontální	0.688803	-0.021701
Vertikální	0.633522	-0.008724
Diagonální	0.473123	0.000665

Tabulka 7.3: Křížová korelace sousedních koeficientů pro G složku

Směr	Původní oblast	Zašifrovaná oblast
Horizontální	0.525467	0.003922
Vertikální	0.466093	-0.005449
Diagonální	0.322522	-0.005692

Tabulka 7.4: Křížová korelace sousedních koeficientů pro B složku

Směr	Původní oblast	Zašifrovaná oblast
Horizontální	0.613084	-0.013722
Vertikální	0.559215	-0.002110
Diagonální	0.361566	0.007187

7.4.3. Výkonnost šifrovacího algoritmu

V následujícím testu byl opět šifrován 24-bitový obraz „Lena“ o velikosti 256x256 pixelů na procesoru AMD Turion 64 X2 s taktovací frekvencí 2.0GHz a operační pamětí 2.0GB RAM. Tabulka 7.5. obsahuje dobu trvání v sekundách pro různé iterace Cliffordova systému a různý počet šifrovacích rund.

Tabulka 7.5: Výkonnost šifrovacího algoritmu při použití waveletové transformace

Iterace / Rundy	1	5	10	15	20	25
1	0.014	0.035	0.105	0.110	0.187	0.212
2	0.037	0.114	0.120	0.254	0.291	0.341
3	0.061	0.174	0.238	0.309	0.401	0.495
4	0.082	0.189	0.281	0.397	0.492	0.607
5	0.098	0.244	0.353	0.464	0.617	0.779

Oproti klasickému šifrovacímu algoritmu byly výpočty zrychleny průměrně 16-krát. Maximální rychlost zpracování obrazových dat nyní činí 13.39MB/s a při dostatečném zabezpečení zašifrovaného obrazu, kdy je nutné použít aspoň 5 iterací Cliffordova systému a 5 šifrovacích rund, se rychlost zpracování obrazových dat sníží na 786kB/s. Výpočetní čas byl tedy výrazně snížen právě díky výběru důležitých waveletových koeficientů. Přitom bezpečnost zašifrovaného obrazu stále zůstává dostatečná. Nicméně, při zanedbání koeficientů, reprezentujících detaily obrazu, dochází při rekonstrukci ke ztrátě informace. Tento šifrovací algoritmus lze tedy použít v aplikacích pracujících v reálném čase, kde požadavky na kvalitu jsou druhořadé.

7.4.4. Ztrátová komprese

Protože kryptografický systém pracuje pouze s waveletovými koeficienty submatice C_0, značně se tím omezí množství dat, která jsou potřeba pro zpětné dešifrování a rekonstrukci obrazu. Přestože ve výše uvedených testech se využívala pouze šestnáctina waveletové oblasti, neznamená to, že po šifrování je nutné uložit pouze šestnáctinu dat. Pixely obrazu jsou reprezentovány bytovými hodnotami v rozmezí 0-255. Naproti tomu waveletové koeficienty nabývají větších rozsahů hodnot a v případě použití komplexních waveletových funkcí i hodnot neceločíselných. Tyto waveletové koeficienty je nutné pro pozdější dešifrování uložit.

V případě Haarovy báze lze waveletový koeficient vyjádřit 2 byty. Pro rastrový obraz s 24-bitovou hloubkou a velikostí 256x256 pixelů je zapotřebí

192kB dat. Pokud se obraz dyadicky dekomponuje a zachová se pouze subpole C_0 (tedy šestnáctina waveletové oblasti), získáme tak dispozici 64x64 waveletových koeficientů, které reprezentují pro 3 barevné hloubky celkem 24kB dat. Kompresní poměr mezi původním obrazem a získanou waveletovou oblastí tedy není 16:1, jak by se na první pohled mohlo zdát, ale pouze 8:1. I přesto ovšem jde o nezanedbatelný kompresní poměr.

8. KRYPTOANALÝZA POMOCÍ EVOLUČNÍCH ALGORITMŮ

8.1. Diferenciální evoluce

Diferenciální evoluce je jednoduchý, přesto efektivný evoluční algoritmus, který byl v roce 1995 představen Stornem a Pricem [1]. Jeho hlavním úkolem je heuristicky nalézt globální minimum u multimodálních funkcí. Experimenty z mnoha aplikací ukazují, že tento evoluční algoritmus konverguje často rychleji než jiné stochastické algoritmy.

Diferenciální evoluce vytváří novou populaci tak, že pro každého jedince ze staré populace vytvoří jeho potenciálního konkurenta. Do nové populace se pak zařadí jedinec s nižší hodnotou účelové funkce. Algoritmus lze zapsat následujícím pseudokódem:

Inicializace populace (vygenerování N jedinců náhodným způsobem)
While (podmínka běhu, např. počet generací G)
For $i = 1$ **to** N **do**
generace šumového vektoru u
vytvoření konkurenta x' křížením vektoru u a jedince x_i
if $f(x') < f(x_i)$ **then** do populace se zařadí x'
else do populace se zařadí x_i
EndFor
EndWhile

kde N je počet jedinců v populaci, f je účelová funkce a x_i je i-tý jedinec s D parametry.

Generace šumového vektoru u je dána na základě vybrané strategie, která využívá mutační konstantu $F \in (0,2)$. Mezi nejpoužívanější strategie patří DERand1Bin (8.1), DERand2Bin (8.2), DEBest1Bin (8.3) nebo DEBest2Bin (8.4).

$$u = x_1 + F(x_2 - x_3) \tag{8.1}$$

$$u = x_5 + F(x_1 + x_2 - x_3 - x_4) \tag{8.2}$$

$$u = x_{best} + F(x_1 - x_2) \tag{8.3}$$

$$u = x_{best} + F(x_1 + x_2 - x_3 - x_4) \tag{8.4}$$

Křížení šumového vektoru a jedince spočívá ve vygenerování náhodného čísla $R_j \in (0,1)$ a porovnání tohoto čísla s křížící konstantou $C \in (0,1)$. Potencionálního konkurenta *x'* k jedinci x_i lze získat na základě (8.5)

$$x'_j = \begin{cases} u_j & R_j \le C \text{ nebo } j \equiv I \\ x_{i,j} & otherwise \end{cases} \tag{8.5}$$

kde j indexuje parametr jedince a I je náhodně zvolené číslo z intervalu $\{1,2,...,D\}$.

Konstanty F, C a N jsou nastaveny před první inicializací populace.

8.2. Kryptoanalýza

8.2.1. Hledání řídicího parametru na základě podobnosti obrazů

Tato kapitola zkoumá využití různých verzí diferenciální evoluce pro nalezení klíče k neoprávněnému dešifrování.

Pro následující testy předpokládejme, že kryptoanalytik má k dispozici:

- původní obraz A_1 a zašifrovaný obraz B_1
- zašifrovaný obraz B_2, který byl vytvořen stejnou sadou klíčů K jako B_1
- sadu klíčů K, kterou zná. Pouze u jednoho parametru $k_2 \in K$ si není jistý jeho přesným vyjádřením.

Úkolem diferenciální evoluce je tedy nalézt parametr k_2 tak, že kryptoanalytik bude moci dešifrovat obraz B_2 a získat tak jeho původní formu A_2. V takovém případě lze použít jednoduchý postup. Útokem hrubou silou by se procházelo známé okolí parametru k_2 a na základě kompletní sady klíčů K by se dešifroval obraz B_1. V případě, že by dešifrovaný obraz byl totožný s obrazem A_1, pak parametr k_2 by byl správný. Pro účel porovnání dešifrovaného obrazu s obrazem A_1 by se použila křížová korelace, kde maximální korelace znamená totožnost obou obrazů. Útok hrubou silou je

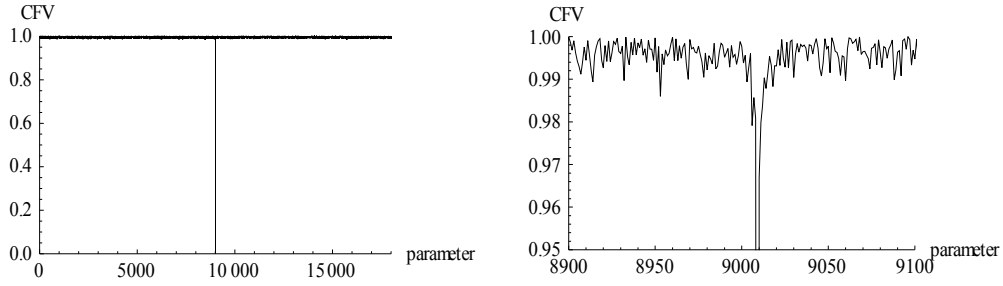
ovšem velmi neefektivní, proto pro účely hledání parametru použijeme diferenciální evoluci.

Víme, že správná hodnota parametru k_2 se pohybuje v intervalu (1.47999999999, 1.48000000001). Z hlediska přesnosti klasického procesoru je nutné tento interval procházet po kroku $1x10^{-15}$. Pro vyjádření podobnosti mezi A_1 a dešifrovaným obrazem je tedy zapotřebí spočítat přibližně $2x10^4$ korelačních koeficientů.

Úkolem diferenciální evoluce je tedy nalézt parametr k_2, který zajistí co nejvyšší korelační koeficient, tedy co nejvyšší podobnost dešifrovaného obrazu a obrazu A_1. Z tohoto důvodu lze hodnotu účelové funkce velmi jednoduše zapsat jako (8.6)

$$CFV = 1 - |correlation\ value| \tag{8.6}$$

Diferenciální evoluce se bude snažit nalézt nulovou hodnotu účelové funkce, která je globálním minimem. Obrázek 8.1 vykresluje účelovou funkci a přiblíženou oblast globálního minima. Na první pohled je patrné, že křivka obsahuje velké množství lokálních extrémů a že zde neexistuje žádný dlouhodobý trend, který by vedl ke globálnímu minimu.



(a) (b)

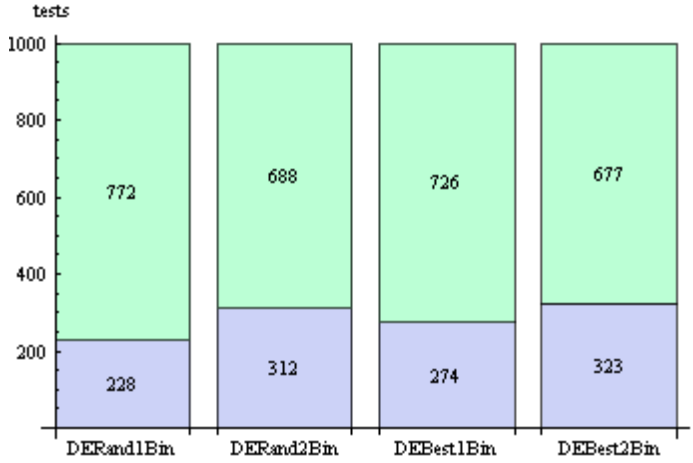
Obrázek 8.1: (a) Účelová funkce, (b) detail v oblasti globálního minima

Algoritmus diferenciální evoluce byl nastaven následovně:

- velikost populace $N = 150$
- počet generací $G = 150$
- mutační konstanta $F = 0.9$
- křížící konstanta $C = 0.3$

- dimenze jedince $D = 1$

Pro zachování objektivity bylo pro různé verze provedeno 1000 testů. Na následujícím Obrázku 8.2 je v horní části sloupců uveden počet neúspěšného hledání, v dolní části počet úspěšných nalezení hledaného parametru.



Obrázek 8.2: Počet úspěšných/neúspěšných pokusů o nalezení parametru

Verze DERand1Bin tedy najde parametr pouze ve 228 případech z 1000, verze DERand2Bin ve 312 případech. Situace je velmi obdobná i u verzí DEBest1Bin a DEBest2Bin, kdy byl správný parametr nalezen ve 274 a 323 případech. Znamená to tedy, že pro výše zmíněné nastavení evolučního algoritmu je pravděpodobnost nalezení správného parametru 22.8%, 31.2%, 27.4% a 32.3% pro používané verze diferenciální evoluce. Maximální počet ohodnocení účelové funkce přesahuje počet korelačních koeficientů.

Pokud budeme srovnávat výkon diferenciální evoluce s útokem hrubou silou, kdy se prochází celý daný prostor, dokud není nalezena správná hodnota, zjistíme několik zajímavých faktů. Je jisté, že při hrubé síle bude počet pokusů o nalezení správného parametru vždy menší nebo roven celkovému počtu korelačních koeficientů a pravděpodobnost nalezení parametru je vždy 100%. Útok hrubou silou může být implementován i jako algoritmus náhodného hledání, kdy se náhodným způsobem vybírá hodnota parametru a poté se zjišťuje, zda je tato hodnota správná. Následující Tabulka 8.1 uvádí srovnání mezi všemi verzemi diferenciální evoluce a algoritmem náhodného hledání. Uvedené hodnoty jsou získány pouze z úspěšných testů.

Tabulka 8.1: Počet ohodnocení účelové funkce/pokusů o nalezení

Algoritmus	Minimální	Průměrný	Zmenšení prostoru
DERand1Bin	1652	12378	38.11%
DERand2Bin	1251	97582	48.29%
DEBest1Bin	1546	13948	41.36%
DEBest2Bin	768	9346	53.27%
Náhodné hledání	837	9057	54.71%

Z prezentovaných dat je zřejmé, že všechny způsoby zmenší prostor prohledávání přibližně na polovinu. Nejlepší výsledky v tomto případě dává algoritmus náhodného hledání, ovšem tvrzení, že se jedná o nejlepší způsob hledání parametru, je velmi sporné, protože pro generování náhodných pozic byl použit pseudonáhodný generátor a na rozdíl od evolučních algoritmů není tento typ hledání založen na heuristice. V případě nasazení jiných typů evolučních algoritmů, jakým jsou genetické algoritmy, simulované žíhání nebo evoluční strategie lze očekávat zvýšení pravděpodobnosti nalezení globálního minima, případně rychlejší konvergenci.

Bylo tedy dokázáno, že evoluční algoritmy jsou schopny nalézt správnou hodnotu parametru k_2 v jeho blízkém okolí na ploše, která nevykazuje žádný dlouhodobý trend. Nicméně využitelnost těchto algoritmů pro účely kryptoanalýzy není jistá, pokud bereme v potaz pravděpodobnost, s jakou jsou evoluční techniky schopny parametr nalézt, a velikost prohledávané plochy v případě, kdy kryptoanalytik vůbec nezná parametr nebo jich nezná hned několik. Tato situace vyústí ve více-rozměrný problém, který je velmi obtížně řešitelný.

8.2.2. Překonání kvantizační jednotky

Kvantizační jednotky v kryptografických systémech jsou velmi jednoduchým, přesto vysoce efektivním způsobem, jak skrýt některé informace. Chaotický systém produkuje reálné výstupy a tyto výstupy lze ořezat na celá čísla, která reprezentují indexy intervalů, ve kterých se výstupy

nacházejí. Taková ztráta informace zapříčiní nemožnost zpětné rekonstrukce orbity systému. Tato kapitola zkoumá způsoby, jak získat počáteční nastavení systému tak, aby generoval výstupy, které známe. Pro odhalování nastavení systému je opět použita diferenciální evoluce.

Daný problém velmi zjednodušíme a pro účely testů použijeme logistickou mapu ve tvaru (8.7).

$$x_{i+1} = rx_i(1 - x_i) \tag{8.7}$$

Výstupy logistické mapy jsou zpracovány kvantizační jednotkou podle (8.8)

$$k_i = (x_i \cdot U)(\mathrm{mod}\, 256) \tag{8.8}$$

kde U je zesílení vstupní hodnoty a x_i je výstup logistické mapy v i-té iteraci.

Získáme tak postupně vektor hodnot $k_1, k_2, ..., k_n$, který můžeme nazývat proud klíče. Tyto hodnoty se aplikují na zprávu $p_1, p_2, ..., p_n$ tak, že se provede XOR operace podle (8.9) a získá se tak posloupnost zašifrované zprávy $c_1, c_2, ..., c_n$.

$$c_i = p_i \oplus k_i \tag{8.9}$$

kde $i = 1,2,...n$.

Lze tedy vidět, že tento způsob šifrování je rozdílný a velmi jednoduchý ve srovnání s kryptografickým systémem navrženém v této práci. Přesto je pro následující testy více než dostačujícím.

Zpráva $p_1, p_2, ..., p_n$ délky n je zašifrována klíčem $k_1, k_2, ..., k_n$. Nyní předpokládejme, že kryptoanalytik má k dispozici část původní zprávy a odpovídající část zašifrované zprávy. Tato část je délky m, kde $m << n$. Kryptoanalytik může tedy analyzovat rozdíly mezi oběma částmi dat a pokusit se o odhalení skrytých informací. Tato metoda se nazývá „known-plaintext attack“. Kryptoanalytik použije získané části zprávy $p_1, p_2, ..., p_m$ a příslušnou část zašifrované zprávy $c_1, c_2, ..., c_m$ pro rekonstrukci proudu klíče $k_1, k_2, ..., k_m$ na základě jednoduché inverzní XOR operace (8.10)

$$k_j = p_j \oplus c_j \tag{8.10}$$

kde $j = 1,2,...m$.

Nyní je tedy k dispozici část proudu klíče, která modifikuje část původní zprávy na část zašifrované zprávy. Kryptoanalytik se nyní může pokusit o získání zbytku klíče $k_{m+1}, k_{m+2}, ..., k_n$. Teprve poté bude schopen dešifrovat zbývající zašifrovanou zprávu $c_{m+1}, c_{m+2}, ..., c_n$. Chaotické systémy ovšem nelze jednoduše predikovat, proto nejlepším způsobem, jak získat proud klíče, je odhalit počáteční nastavení systému. K tomuto bude využita diferenciální evoluce.

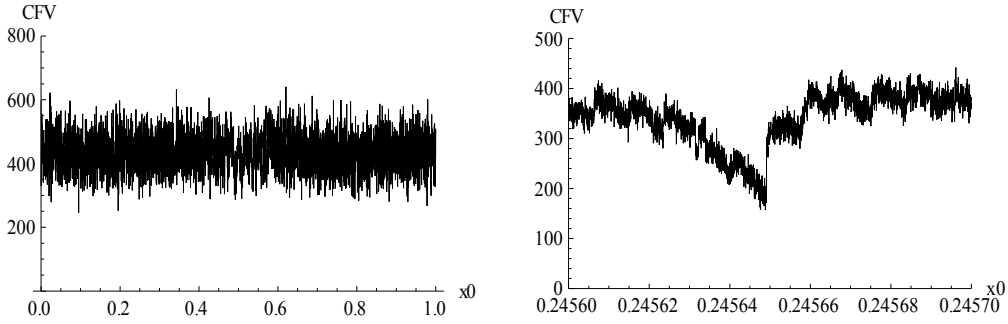
Jedno-rozměrný problém

V prvním testu předpokládáme, že známe část výstupu kvantitní jednotky, tzn. proud klíče $k_1, k_2, ..., k_m$ o délce $m = 100$, známe řídicí parametr r logistické mapy (8.7) a neznáme pouze jeho počáteční podmínku x_0. Úkolem diferenciální evoluce bude tedy odhalit počáteční podmínku tak, aby získaná část proudu klíče $k_1, k_2, ..., k_m$ byla stejná jako výstupy produkované logistickou mapou a kvantizační jednotkou, $e_1, e_2, ..., e_m$.

Účelová funkce byla vytvořena jako váhovaný rozdíl mezi $k_1, k_2, ..., k_m$ a $e_1, e_2, ..., e_m$ podle (8.11).

$$CF = \sum_{i=1}^{m} \begin{cases} |k_i - e_i| / i & k_i \neq e_i \\ 0 & otherwise \end{cases} \tag{8.11}$$

Váha i má za úkol penalizovat chyby v $e_1, e_2, ..., e_m$ s ohledem na správnost chaotické trajektorie od jeho počáteční pozice. Evoluční algoritmus tedy velmi rychle zahodí počáteční podmínky, které způsobují již na začátku velmi rozdílné trajektorie od trajektorie hledané. Následující Obrázek 8.3 zobrazuje účelovou funkci. Přestože není globální minimum z důvodu vykreslování po krocích zřejmé, nulová hodnota účelové funkce se nachází na pozici $\mathrm{x}_0 = 0.2456485653$.



(a) (b)

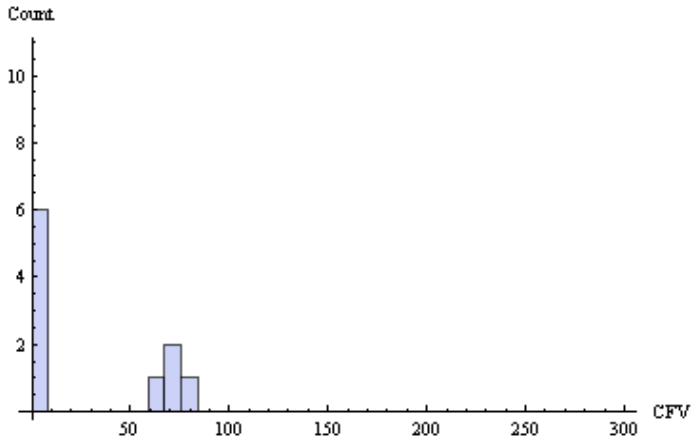
Obrázek 8.3: (a) Účelová funkce pro x_0=0.2456485653, (b) detail v oblasti globálního minima

Algoritmus diferenciální evoluce byl nastaven následovně:

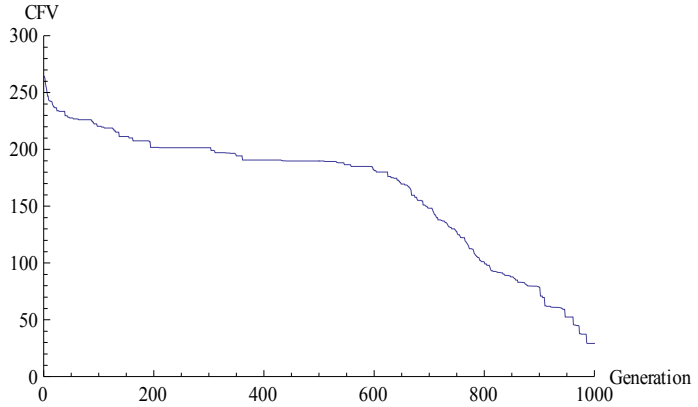
- velikost populace $N = 500$
- počet generací $G = 1000$
- mutační konstanta $F = 0.9$
- křížící konstanta $C = 0.3$
- dimenze jedince $D = 1$

Vzhledem k tomu, že počáteční podmínka je reprezentována jako 64-bitové číslo, pak lze velmi jednoduše spočítat, že bude prováděno 1.0x10E05 ohodnocení účelové funkce na ploše velké 1.8x10E19 . Interval hodnot, ve kterém se počáteční podmínka hledá je nastaven na $(0,1)$.

Obrázek 8.4 ukazuje, že během 10 běhů byla schopna diferenciální evoluce DERand1Bin nalézt správnou počáteční podmínku v 6 případech z 10. V 6 případech byla tedy hodnota účelové funkce rovna nule. Obrázek 8.5 zobrazuje průměrný vývoj konvergence 10 běhů diferenciální evoluce k nejlepší hodnotě účelové funkce vzhledem k počtu provedených generací.

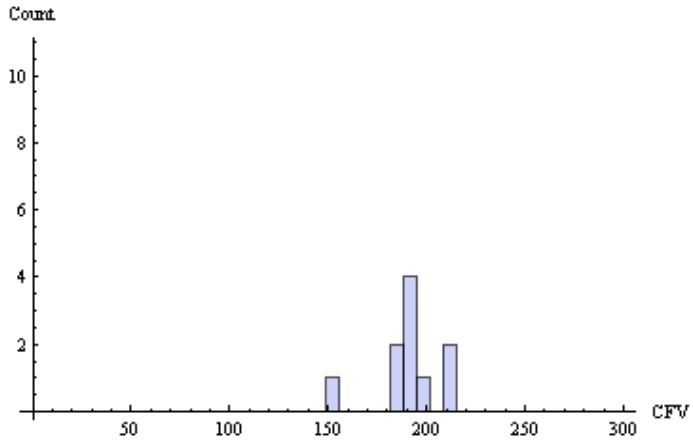


Obrázek 8.4: Histogram výkonu 10 běhů DERand1Bin pro x_0=0.2456485653

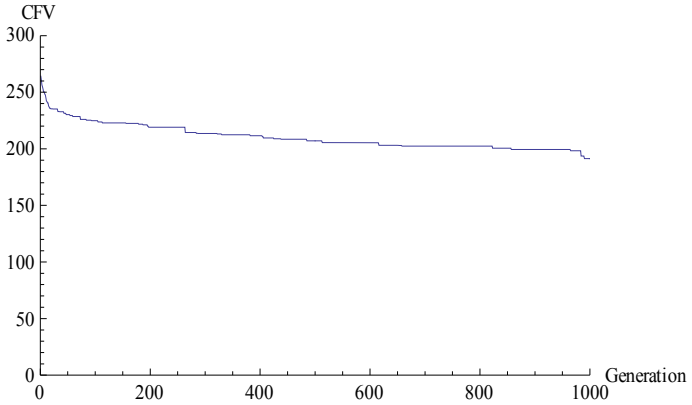


Obrázek 8.5: Konvergence DERand1Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653

Verze DERand2Bin má oproti DERand1Bin mnohem horší výsledky. Obrázek 8.6 ukazuje, že ani během 10 běhů nebyla schopna diferenciální evoluce DERand2Bin nalézt správnou počáteční podmínku a dokonce se k ní ani nepřiblížila. Obrázek 8.7 zobrazuje průměrný vývoj konvergence 10 běhů diferenciální evoluce.

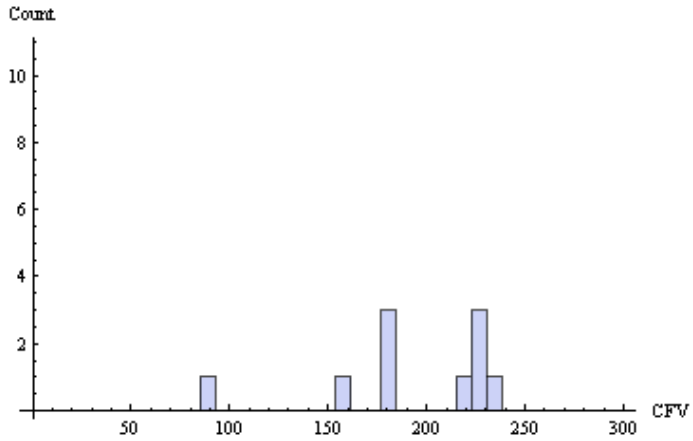


Obrázek 8.6: Histogram výkonu 10 běhů DERand2Bin pro x_0=0.2456485653

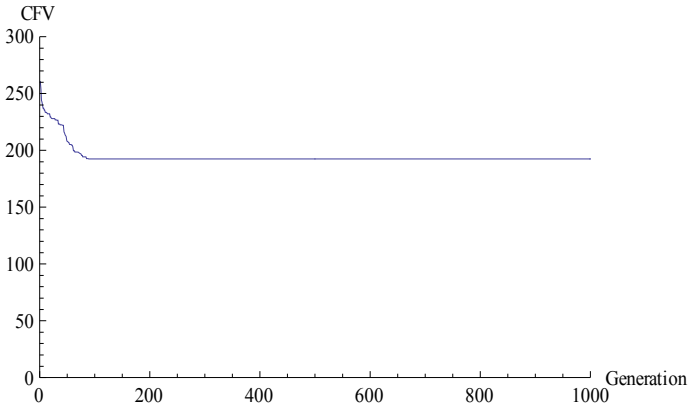


Obrázek 8.7: Konvergence DERand2Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653

Výkon a konvergence diferenciální evoluce DERand1Best je zobrazena na Obrázku 8.8 a Obrázku 8.9. Ani zde nedokázala diferenciální evoluce odhalit počáteční podmínku.

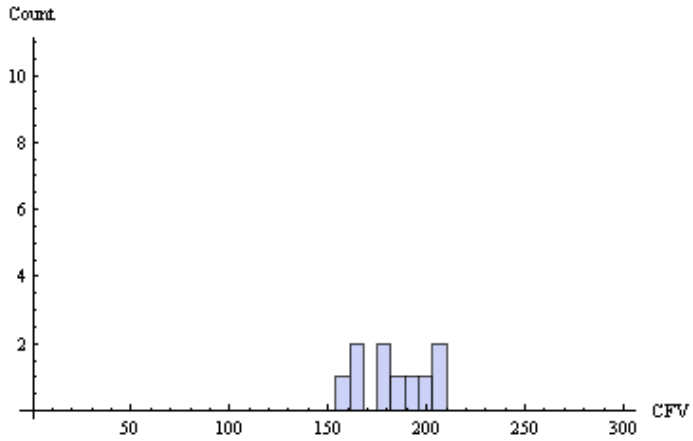


Obrázek 8.8: Histogram výkonu 10 běhů DEBest1Bin pro x_0=0.2456485653

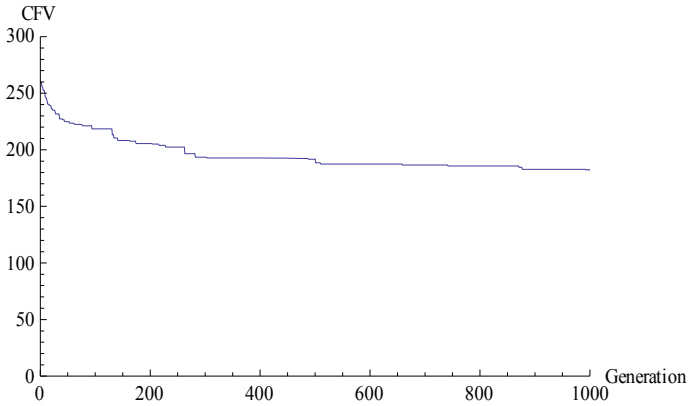


Obrázek 8.9: Konvergence DEBest1Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653

Experimentálně byla také vyzkoušena verze DEBest2Bin, která ovšem také nepodala takový výkon, jako v případě DERand1Bin. Obrázek 8.10 a Obrázek 8.11 ukazují vývoj hodnoty účelové funkce v průběhu generací.



Obrázek 8.10: Histogram výkonu 10 běhů DEBest2Bin pro x_0=0.2456485653



Obrázek 8.11: Konvergence DEBest2Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653

Následující Tabulka 8.2 sumarizuje úspěšnost různých verzí diferenciální evoluce, obsahuje původní a nalezené hodnoty a odchylky mezi nimi. Za nalezenou hodnotu se považuje ta hodnota, která ze všech 10 běhů různých verzí diferenciální evoluce byla nejblíže hledané hodnoty.

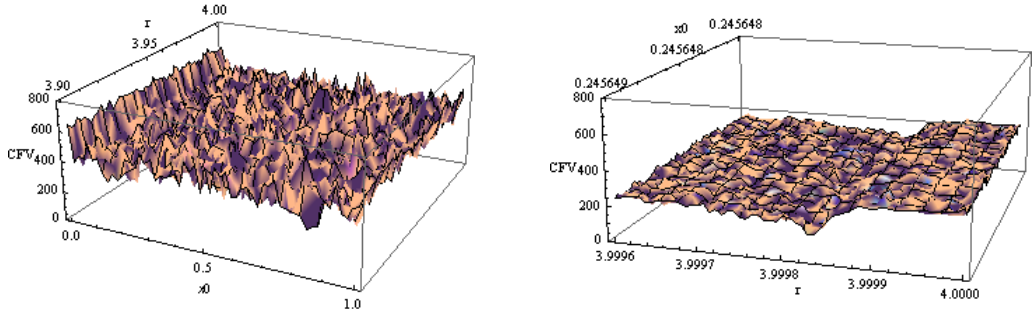
Tabulka 8.2: Odchylky nalezených hodnot pro jedno-rozměrný problém

	DERand1Bin	**DERand2Bin**
Hledaný parametr	$\mathbf{x_0}$	$\mathbf{x_0}$
Původní hodnota	0.2456485653	0.2456485653
Nalezená hodnota	0.2456485653	0.2456485627
Odchylka	0	0.0000000026
	DeBest1Bin	**DEBest2Bin**
Hledaný parametr	$\mathbf{x_0}$	$\mathbf{x_0}$
Původní hodnota	0.2456485653	0.2456485653
Nalezená hodnota	0.2456485472	0.2456489210
Odchylka	0.0000000181	0.0000000645

Verze DERand2Bin, DEBest1Bin a DEBest2Bin nalezly lokální minimum v bodech, které jsou blízko hledané hodnoty. Přesto od původní hledané počáteční podmínky je zde určitá odchylka. Pokud tyto nalezené hodnoty nastavíme jako počáteční podmínky do systému, který generuje proud klíče $e_1, e_2, ..., e_m$, pak mezi proudy $k_1, k_2, ..., k_m$ a $e_1, e_2, ..., e_m$ je pro DERand2Bin rozdílných 88% hodnot, pro DEBest1Bin 91% hodnot a pro DEBest2Bin 97% hodnot.

Dvou-rozměrný problém

Dvourozměrný problém spočívá v neznalosti jak řídicího parametru r, tak počáteční podmínky x_0 logistické mapy (8.7). K dispozici máme pouze proud klíče $k_1, k_2, ..., k_m$ o délce $m = 100$, Účelová funkce je totožná s účelovou funkcí jedno-rozměrného problému (8.11). Obrázek 8.12 vykresluje účelovou funkci jako dvou-rozměrnou plochu.



(a) (b)

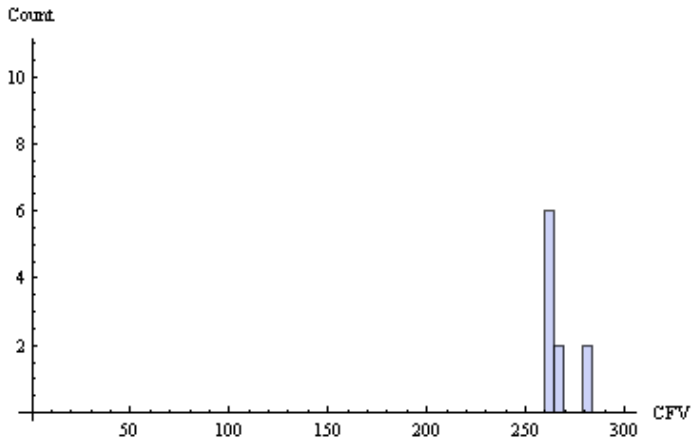
Obrázek 8.12: (a) Účelová funkce pro x_0=0.2456485653 a r=3.9998472, (b) detail v oblasti globálního minima

Algoritmus diferenciální evoluce byl nastaven následovně:

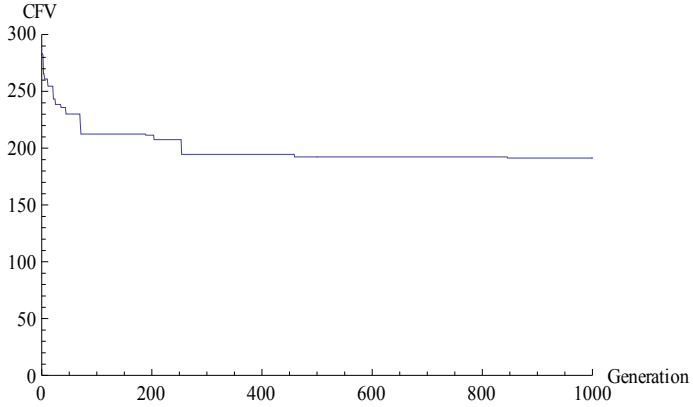
- velikost populace $N = 500$
- počet generací $G = 1000$
- mutační konstanta $F = 0.9$
- křížící konstanta $C = 0.3$
- dimenze jedince $D = 2$

Interval hodnot, ve kterém se počáteční podmínka hledá je nastaven na $(0,1)$ a interval pro řídicí parametr je $(3.99996,4)$.

Obrázek 8.13 ukazuje, že během 10 běhů nebyla diferenciální evoluce DERand1Bin schopna nalézt správnou počáteční podmínku ani řídicí parametr. Obrázek 8.14 zobrazuje průměrný vývoj konvergence 10 běhů diferenciální evoluce k nejlepší hodnotě účelové funkce vzhledem k počtu provedených generací. Lze vidět, že již po 250 generacích není zřetelná žádná výraznější konvergence ke globálnímu minimu.

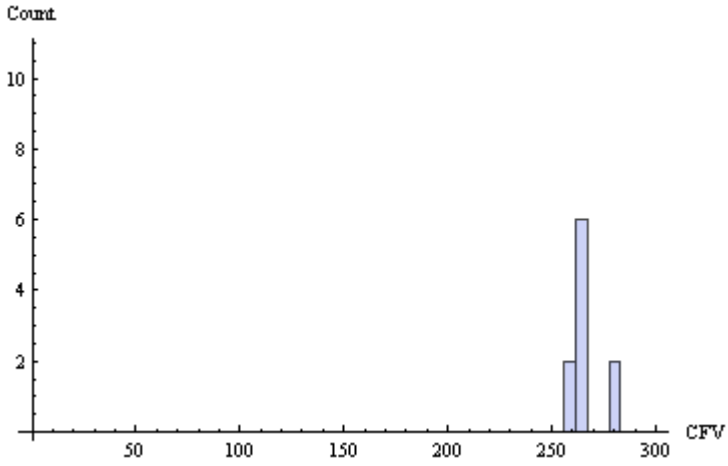


Obrázek 8.13: Histogram výkonu 10 běhů DERand1Bin pro x_0=0.2456485653 a r=3.9998472

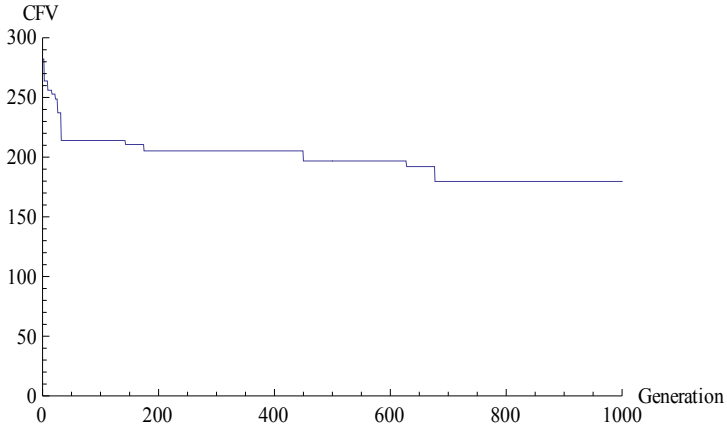


Obrázek 8.14: Konvergence DERand1Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653 a r=3.9998472

Verze DERand2Bin také není schopna během 1000 generací odhalit hledané hodnoty. Obrázky 8.15 a 8.16 ukazují úspěšnost DERand2Bin.

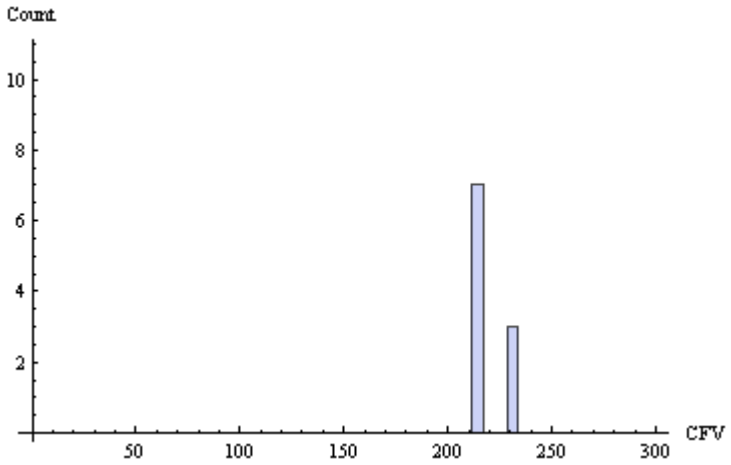


Obrázek 8.15: Histogram výkonu 10 běhů DERand2Bin pro x_0=0.2456485653 a r=3.9998472

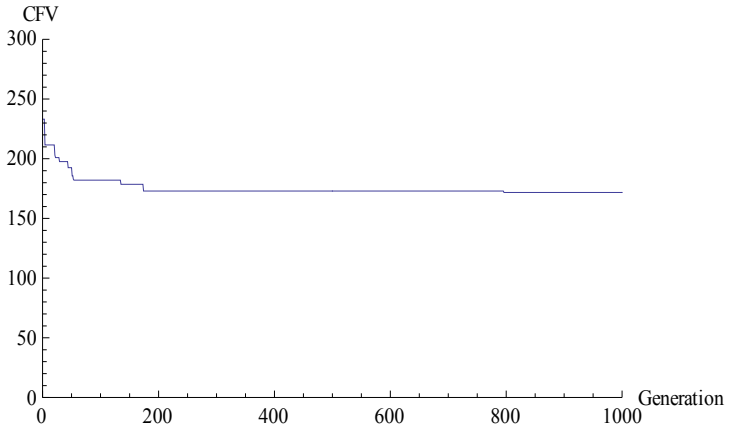


Obrázek 8.16: Konvergence DERand2Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653 a r=3.9998472

Při nasazení verze DEBest1Bin nebylo již po 200 generacích patrné výraznější přiblížení ke globálnímu minimu. Obrázky 8.17 a 8.18 ukazují úspěšnost DEBest1Bin.

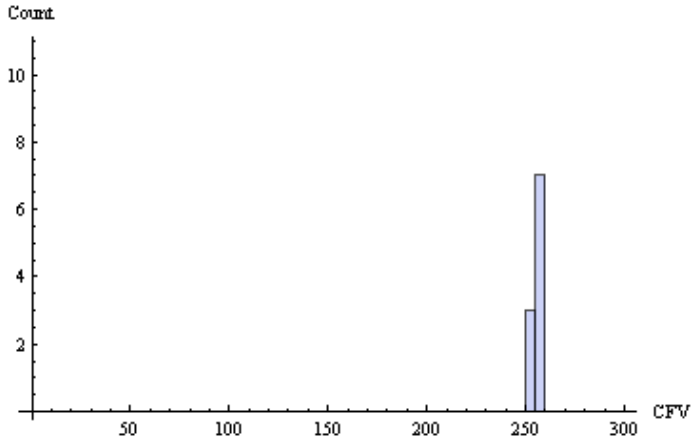


Obrázek 8.17: Histogram výkonu 10 běhů DEBest1Bin pro x_0=0.2456485653 a r=3.9998472

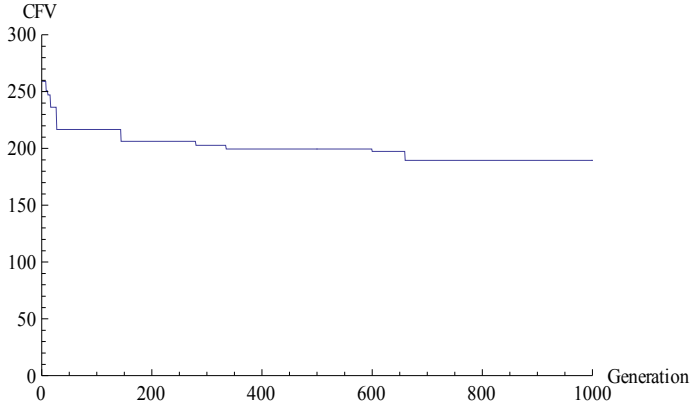


Obrázek 8.18: Konvergence DEBest1Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653 a r=3.9998472

Verze DEBest2Bin má velmi obdobné výsledky jako DERand1Bin. Obrázky 8.19 a 8.20 ukazují úspěšnost DEBest2Bin.



Obrázek 8.19: Histogram výkonu 10 běhů DEBest2Bin pro x_0=0.2456485653 a r=3.9998472



Obrázek 8.20: Konvergence DEBest2Bin k nejlepší hodnotě účelové funkce pro x_0=0.2456485653 a r=3.9998472

Následující Tabulka 8.3 sumarizuje úspěšnost různých verzí diferenciální evoluce, obsahuje původní a nalezené hodnoty a odchylky mezi nimi.

Tabulka 8.3: Odchylky nalezených hodnot pro dvou-rozměrný problém

	DERand1Bin		**DERand2Bin**	
Hledaný parametr	$\mathbf{x_0}$	**r**	$\mathbf{x_0}$	**r**
Původní hodnota	0.2456485653	3.9998472	0.2456485653	3.9998472
Nalezená hodnota	0.2990154983	3.9999864	0.2526396225	3.9999945
Odchylka	0.0533669330	0.0001392	0.0069810572	0.0001473
	DEBest1Bin		**DEBest2Bin**	
Hledaný parametr	$\mathbf{x_0}$	**r**	$\mathbf{x_0}$	**r**
Původní hodnota	0.2456485653	3.9998472	0.2456485653	3.9998472
Nalezená hodnota	0.2392399157	3.9999849	0.2630114485	3.99998137
Odchylka	0.0064086496	0.0001377	0.0173628832	0.00013417

Pokud opět nalezené hodnoty nastavíme jako počáteční podmínky a řídicí parametry do systému, který generuje proud klíče $e_1, e_2, ..., e_m$, pak mezi proudy $k_1, k_2, ..., k_m$ a $e_1, e_2, ..., e_m$ bude rozdílných 100% hodnot. Přestože se může zdát, že odchylka řídicího parametru r je relativně malá, je nutné si uvědomit, že hledání tohoto parametru probíhalo na velmi zúženém intervalu $(3.99996, 4)$. Algoritmus diferenciální evoluce měl tedy upřesněnou oblast, ve které se nachází správný řídicí parametr. Přesto nebyla diferenciální evoluce schopna správné hodnoty nalézt.

Sumarizace poznatků

Vzhledem k tomu, že diferenciální evoluce není schopna nalézt pro více-rozměrný problém správné hodnoty pro nastavení chaotického systému, nelze ani získat zbývající proud klíče $k_{m+1}, k_{m+2}, ..., k_n$, který kryptoanalytik nezná. Ve výše uvedeném případě se jednalo o velmi jednoduchý kryptografický systém s jednou iterativní mapou a dvěma šifrovacími klíči (počáteční podmínka a řídicí parametr). Každá hodnota ve zprávě je navíc šifrována separátně.

Kryptografický systém navržený v této práci využívá celkem čtyři iterativní mapy (8.12), které jsou na sobě závislé.

$$\begin{aligned} x_{n+1} &= \sin(a \cdot y_n) + c \cdot \cos(a \cdot x_n) \\ y_{n+1} &= \sin(b \cdot x_n) + d \cdot \cos(b \cdot y_n) \\ z_{n+1} &= \sin(e \cdot x_n) + f \cdot \cos(e \cdot z_n) \\ w_{n+1} &= \sin(g \cdot y_n) + h \cdot \cos(e \cdot w_n) \end{aligned} \tag{8.12}$$

Každá mapa využívá dva řídicí parametry. Pokud by kryptoanalytik chtěl odhalit, jakým způsobem jsou pixely v obrazu permutovány, musel by znát nastavení tří iterativních map pro pozice x, y, z. To si vyžaduje znalost šesti řídicích parametrů. Pokud by chtěl znát, jakým způsobem jsou pixely modifikovány, musel by opět znát nastavení tří iterativních map, tedy mapu x, jejíž výstup je vstupem mapy y, mapu y, jejíž výstup je vstupem mapy w a mapu w, která se stará o samotnou modifikaci hodnot pixelů. Minimálně se tedy jedná o šesti-rozměrný problém. Jak bylo z výsledků této kapitoly vidět, již dvou-rozměrný problém je pro diferenciální evoluci velmi obtížný. Můžeme tedy říci, že evoluční algoritmy poskytují velice zajímavé metody kryptoanalýzy, ovšem navržený kryptografický systém je bezpečný a pravděpodobně neřešitelný evolučními algoritmy.

9. ZÁVĚR

Tato disertační práce se zabývala návrhem efektivního kryptografického systému pro obrazy. Jádrem celého tohoto systému je nelineární dynamický systém, který vykazuje chaotické chování. Konkrétně se jedná o Cliffordův systém, který je popsán dvěma iterativními mapami. Cliffordův systém byl rozšířen do čtyř-dimenzionální podoby tak, aby mohl být plně použit pro šifrovací účely. První tři iterativní mapy mají za úkol permutovat pixely obrazu nejen v rámci jeho plochy, ale také mezi barevnými rovinami. Čtvrtá iterativní mapa se stará o vygenerování hodnoty, která je použita pro modifikaci příslušného pixelu. Řídicí parametry Cliffordova systému se považují za šifrovací/dešifrovací klíče a pro nastavení počátečních podmínek se používají pozice a hodnoty pixelů. Tento způsob nastavování počátečních podmínek má velmi zajímavý efekt. V případě, že budou šifrována různá data stejnými šifrovacími klíči, pak dojde k vygenerování rozdílných permutačních a modifikačních předpisů.

Detailní analýza bezpečnosti ukázala, že kryptografický systém má velmi dobré difúzní a konfúzní vlastnosti. Dokáže efektivně dekorelovat sousední pixely a zašifrované obrazy mají velmi vysokou míru entropie. V případě minimální odchylky v šifrovacích klíčích získáme rozdílnou zašifrovanou formu obrazu a při minimální odchylce v dešifrovacích klíčích nelze obraz rekonstruovat ani do jeho přibližné podoby. Protože navržený kryptografický systém používá osm řídicích parametrů, které jsou reálného charakteru, prostor klíčů je široký dost na to, aby zabránil útokům hrubou silou.

Hlavní nevýhodou celého systému se ukázala jeho výpočetní náročnost, proto byla použita waveletová transformace pro extrakci důležitých informací z obrazu, přičemž detaily obrazy byly zanedbány. Získané waveletové koeficienty se následně zašifrovaly. Waveletová oblast se změnila natolik, že při rozdílných dešifrovacích klíčích nebylo možné z waveletových koeficientů rekonstruovat aproximaci obrazu. Kvůli vynechání velkého množství koeficientů, reprezentujících detaily obrazu, dochází ke ztrátě informace. Tato ztráta se negativně projevuje pixelizací či jinou distorzí v dešifrovaném obrazu. Nicméně je tím zajištěno nejen urychlení celého šifrovacího procesu, ale také určitá úroveň komprese.

Odolnost navrženého kryptografického systému byla ověřena i proti evolučním algoritmům, což jsou velmi efektivní heuristické a optimalizační metody. Konkrétně byla použita diferenciální evoluce. V prvním testu se prohledávalo okolí hodnoty jednoho řídicího parametru Cliffordova systému a byla snaha získat jeho správnou hodnotu. Diferenciální evoluce tuto správnou hodnotu našla a omezila prostor hledání až o polovinu. Je ovšem nutné si uvědomit, že okolí řídicího parametru bylo známo a že kryptografický systém pracuje celkem s osmi řídicími parametry. Principem druhého testu bylo vytvořit jednoduchý kryptografický systém s kvantizační jednotkou a pomocí diferenciální evoluce se snažit nalézt nastavení chaotického systému. V případě hledání více než jednoho parametru vyústila úloha ve více-rozměrný problém, který diferenciální evoluce nebyla schopna vyřešit. Nastavení chaotického systému tedy nelze odhalit ani evolučními algoritmy.

Hlavní cíle disertační práce byly stanoveny na začátku této práce a jejich splnění může být rozebráno v následujících bodech.

1. **Navrhnout metodu pro šifrování obrazů**
   Byl navržen kryptografický systém, který patří mezi blokové šifry a je speciálně vytvořen pro šifrování obrazů. Permutační operace probíhají po celém obrazu a mezi všemi barevnými rovinami. Kryptografický systém je natolik efektivní, že ze zašifrovaných obrazů nelze vyčíst žádnou konkrétní informaci.

2. **Využít nelineární dynamický systém, který vykazuje chaotické chování**
   Navržený kryptografický systém využívá Cliffordův systém, který generuje podivné atraktory. Cliffordův systém splňuje všechny požadavky chaotického chování - je citlivý na počáteční podmínky a řídicí parametry, je topologicky tranzitivní a má dostatečnou hustotu a délku period.

3. **Provést detailní analýzu zabezpečení**
   Analýza bezpečnosti ukázala, že kryptografický systém je schopen generovat zašifrované formy obrazu s velmi vysokou entropií, velmi

efektivně dekoreluje sousední pixely a je citlivý nejen na změny v šifrovacích/dešifrovacích klíčích, ale také na samotnou změnu informace v obrazu.

4. **Provést srovnání se stávajícími kryptografickými systémy**
   Srovnáme-li kryptografický systém se systémy jinými, které jsou rovněž založeny na deterministickém chaosu, pak navržená metoda má efektivnější dekorelační účinky a mnohem větší prostor klíčů. Vzhledem k tomu, že v navržené metodě i pixely hrají roli při nastavení systému, šifrovací předpisy při stejných šifrovacích klíčích budou pro různé obrazy vždy jiné. Tento způsob se liší od jiných chaotických šifer, u kterých vždy stejné šifrovací klíče vedou ke generování stejných šifrovacích předpisů. Hlavní nevýhodou navrženého systému je jeho výpočetní náročnost.

5. **Prozkoumat možnosti zavedení algoritmů do aplikací pracujících v reálném čase**
   Vzhledem k tomu, že obraz obsahuje velké množství dat a lidské zrakové vnímání je robustní na menší ztrátu nebo degradaci informace, je možné využít ztrátovou kompresi. Pomocí waveletové analýzy a selekce signifikantní informace lze šifrovací proces značně urychlit. I přesto je ovšem nutné v budoucnu provést několik dalších úprav tak, aby bylo možné nasadit navržený kryptografický systém do aplikací, jakým jsou videokonference.

6. **Ověřit rezistenci kryptografického systému proti útokům evolučních algoritmů**
   Experimentálně byly vyzkoušeny některé techniky kryptoanalýzy pomocí evolučních algoritmů, konkrétně diferenciální evoluce a její verze DERand1Bin, DERand2Bin, DEBest1Bin a DEBest2Bin. Bylo prokázáno, že za určitých podmínek lze u jednoduchých šifrovacích algoritmů získat nastavení chaotického systému, který generuje šifrovací předpisy. Nicméně pokud je kryptografický systém složitější, ani evoluční algoritmy pravděpodobně ještě stále nedokážou tajné informace systému odhalit.

LITERATURA

[1] SPROTT, J.C., *Chaos and Time Series Analysis*, Oxford University Press, 2003, ISBN 978-0-19-850840-3

[2] STALLINGS, W., *Cryptography and Network Security: Principles and Practice*, Prentice-Hall, 1999, ISBN 978-0-13-091429-3

[3] KOTULSKI, Z., SZCZEPANSKI, J., *Discrete chaotic cryptography (DCC)*, Nonlinear Evolution Equations and Dynamical Systems – NEEDS'97, 1997

[4] SHANNON, C., *Communication Theory of Secrecy Systems*, Bell System Technical Journal, vol. 28 (4)

[5] LI, S., ZHENG, X., *Cryptanalysis of a Chaotic Image Encryption Method*, The 2002 IEEE International Symposium on Circuits and Systems, Proceedings of ISCAS 2002, vol. 2, pp. 708-711

[6] DEVANEY, R.L., *An Introduction to Chaotic Dynamical Systems*, Westview Press, 1989, ISBN 978-0-81-334085-2

[7] SPROTT, J.C., *Creating Patterns in Chaos*, M&T Books, 1993, ISBN 978-1-55-851298-6

[8] TSIMRING, L., TENNY, R., *Security issues in chaos-based communication and encryption*, Winter school on Chaotic Communications, 2003, Dostupné na: http://inls.ucsd.edu/~lev/ws2003/WS2003_Tsimring.pdf

[9] MAO, Y., CHEN, G., *Chaos-Based Image Encryption*, Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics, Springer Berlin, 2003, ISBN 978-3-540-20595-1

[10] CHUA, L., YAO, Y., YANG, Q., *Generating randomness from chaos and constructing chaos with desired randomness*, International Journal of Circuit Theory and Application, 1990, vol. 18, pp. 215–240, ISSN 0098-9886

[11] MATHEWS, R., *On the derivation of a ‚chaotic' encryption algorithm*, Cryptologica, 1989, vol. 13, is. 1, pp. 29-42, ISSN 0161-1194

[12] WHEELER, D.D., *Problems with Chaotic Cryptosystems*, Cryptologia, 1989, vol. 12, is. 3, pp. 243-250

[13] BIANCO, M.E., REED, D.A., *Encryption System Based on Chaos Theory*, US Patent No. 5,048,086, Sep. 10., 1991

[14] PROTOPOPESCU, V.A., SANTORO, R.T., TOLLIVER, J.S., *Fast and Secure Encryption- Decryption Method Based on Chaotic Dynamics*, US Patent No. 5,479,513, Dec. 26., 1995

[15] DEFFEYES, K.S., *Encryption System and Method*, US Patent No. 5,001,754, Mar.19., 1991

[16] GUTOWICZ, H.A., *Cryptography with Dynamical Systems*, Cellular Automata and Cooperative Systems, 1993, pp. 237–274

[17] FRIDRICH, J., *Symmetric ciphers based on two-dimensional chaotic maps*, International Journal of Bifurcation and Chaos, 1998, vol. 8, is. 6, pp. 1259-1284

[18] YEN, J.C., GUO, J.I., *A new chaotic key-based design for image encryption and decryption*, Proceedings of the IEEE International Symposium Circuits and Systems, Geneva, Switzerland, 2000, vol. 4, pp. 49-52, ISBN 0-7803-5482-6

[19] FU, CH., ZHANG, Z., CHEN, Z., WANG, X., *An Improved Chaos-Based Image Encryption Scheme*, ICCS 2007, Proceedings of the 7th international conference on Computational Science, Beijing, China, 2007, pp. 575-582, ISBN 978-3-540-72583-1

[20] CHEN, G., MAO, Y.B., CHUI, C.K., *A symmetric image encryption scheme based on 3D chaotic cat maps*, Chaos, Solitons & Fractals, 2004, vol. 12, is. 3, pp. 749-761, ISSN 0960-0779

[21] MAO, Y.B., CHEN, G., LIAN, S.G., *A novel fast image encryption scheme based on the 3D chaotic baker map*, International Journal of Bifurcation and Chaos, 2004, vol. 14, is. 10, pp. 3163-3624, ISSN 0218-1274

[22] LIAN, S.G. SUN, J., WANG, Z., *A block cipher based on a suitable use of chaotic standard map*, Chaos, Solitons & Fractals, 2005, vol. 26, is. 1, pp. 117-129, ISSN 0960-0779

[23] WONG, K.W., KWOK, S.H., LAW, W.S., *A Fast Image Encryption Scheme based on Chaotic Standard Map*, Physics Letters A, 2006, vol. 372, is. 15, pp. 2645-2652, ISSN 0375-9601

[24] HE, X., ZHU, Q., GU, P., *A New Chaos-Based Encryption Method for Color Image*, Rough Sets and Knowledge Technology, Springer Berlin, 2006, pp. 671-678, ISBN 978-3-540-36297-5

[25] AHMED, H., KALASH, H., ALLAH, O., *An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption*, Informatica, 2007, vol. 31, is. 1, pp. 121-129, ISSN 0350-5596

[26] LIAN, S., SUN, J., WANG, Z., *Security analysis of a chaos-based image encryption algorithm*, Physica A, 2005, vol. 351, pp. 645-661, ISSN 0378-4371

[27] KELBER, K., SCHWARTZ, W., *General Design Rules for Chaos-Based Encryption Systems*, NOLTA 2005, Bruges

[28] ZHAO, X-Y., CHEN, G., ZHANG, D., WANG, X-H., DONG, G-C., *Decryption of pure-position permutation algorithms*, Journal of Zhejiang University SCIENCE, 2004, vol. 5, is. 7, pp. 803–809

[29] GAO, T., CHEN, Z., *A new image encryption based on hyper-chaos*, Physics Letters A, 2008, vol. 372, is. 4, pp. 394-400, ISSN 0375-9601

[30] FU, CH., ZHANG, Z., CHEN, Z., WANG, X., *An Improved Chaos-Based Image Encryption Scheme*, ICCS 2007, Springer-Verlag, Berlin, 2007

[31] GAO, H.J., ZHANG, Y.S., LIANG, S.Y., *A new chaotic algorithm for image encryption*, Chaos, Solitons &Fractals, 2006, vol. 29, pp. 393-399, ISSN 0960-0779

[32] CHUANMU, L., LIANXI, H., *A new image encryption scheme based on hyperchaotic sequences*, IEEE International Workshop on Anti-counterfeiting, Security, Identification, Xiamen, Fujian, 2007, pp. 237-240

[33] WANG, L., YE, Q., XIAO, Y., ZOU, Y., ZHANG, B., *An Image Encryption Scheme Based on Cross Chaotic Map*, Proceedings of the 2008 Congress on Image and Signal Processing, 2008, vol. 3, pp. 22-26, ISBN 978-0-7695-3119-9

[34] ZHAI, Y., LIN, S., ZHANG, Q., *Improving Image Encryption Using Multi-chaotic Map*, Workshop on Power Electronics and Intelligent Transportationi System, 2008, pp. 143-148, ISBN 978-0-7695-3342-1

[35] SHENG, Y., *Wavelet Transform*, The Transforms and Applications, Handbook, CRC Press LLC, 2002, ISBN 978-0849383427

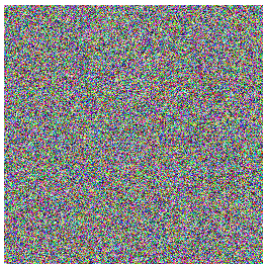
[36] VALENS, C., *A Really Friendly Guide toWavelets*, Dostupné na: http://pagesperso-orange.fr/polyvalens/clemens/download/arfgtw.pdf

[37] STORN, R., PRICE, K., *Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report TR-95-012, ICSI, 1995.

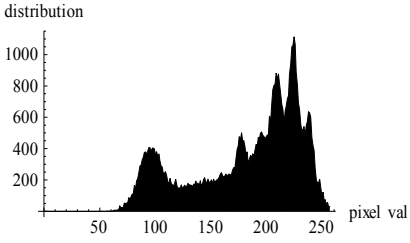
10. PŘÍLOHA

10.1. Obraz „Lena“

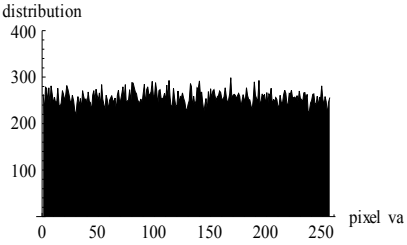
Původní obraz A



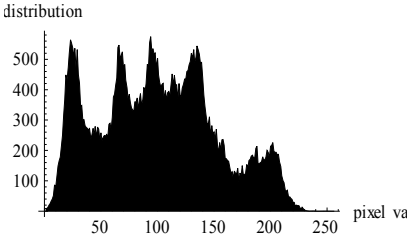
Zašifrovaný obraz B



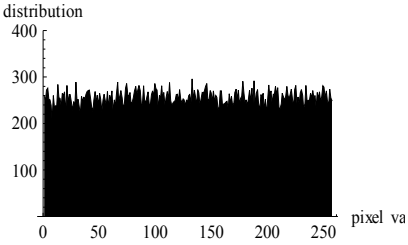
Distribuce R složky obrazu A



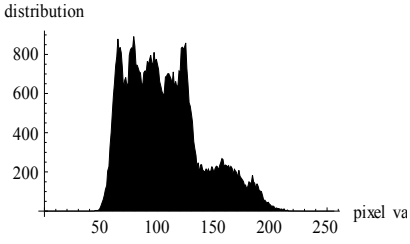
Distribuce R složky obrazu B



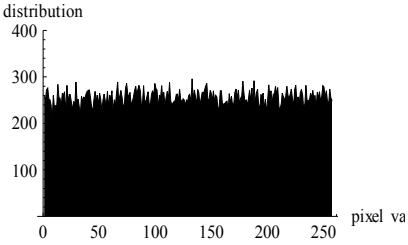
Distribuce G složky obrazu A



Distribuce G složky obrazu B



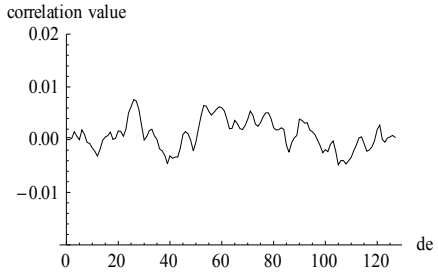
Distribuce B složky obrazu A



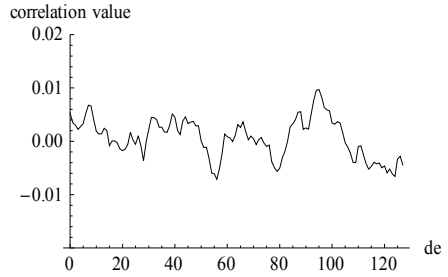
Distribuce B složky obrazu B

Entropie obrazů

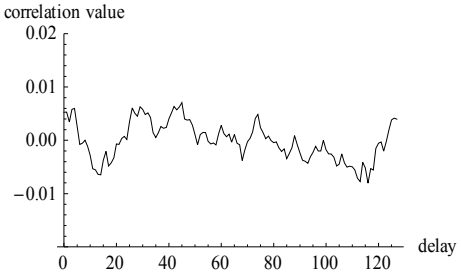
RGB	Původní obraz	Zašifrovaný obraz
R	7.25086	7.99742
G	7.59057	7.99730
B	6.92842	7.99689



Křížová korelace R složek



Křížová korelace G složek



Křížová korelace B složek

Křížová korelace sousedních pixelů pro R složku

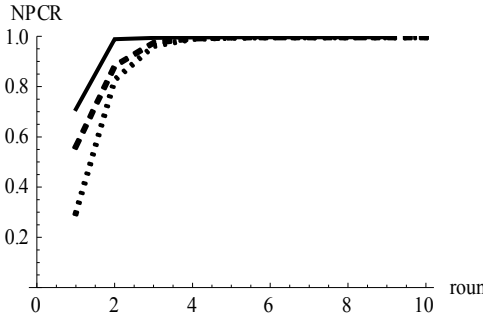
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.954094	0.002585
Vertikální	0.976929	-0.002036
Diagonální	0.929461	0.000506

Křížová korelace sousedních pixelů pro G složku

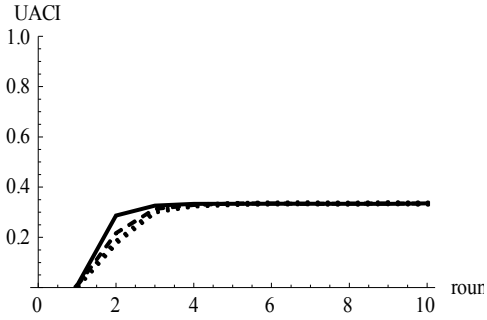
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.938597	-0.003592
Vertikální	0.968472	-0.001187
Diagonální	0.913181	0.006435

Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.922301	0.000719
Vertikální	0.951445	0.000225
Diagonální	0.892751	-0.004336



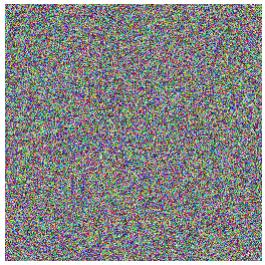
Vývoj NPCR pro šifrovací rundy



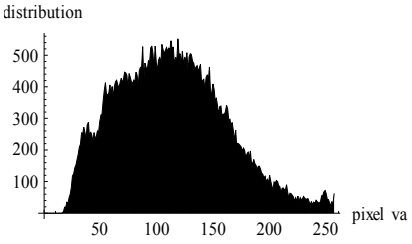
Vývoj UACI pro šifrovací rundy

10.2. Obraz „Flowers“

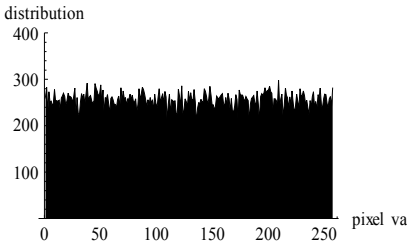
Původní obraz A



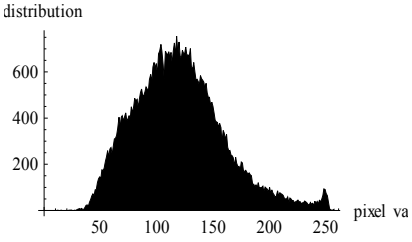
Zašifrovaný obraz B



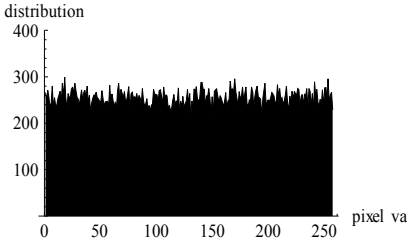
Distribuce R složky obrazu A



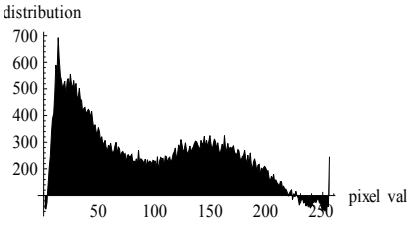
Distribuce R složky obrazu B



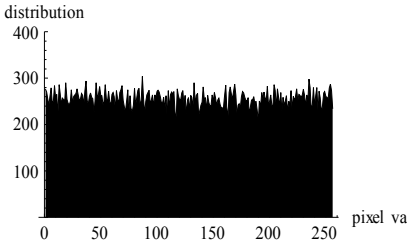
Distribuce G složky obrazu A



Distribuce G složky obrazu B



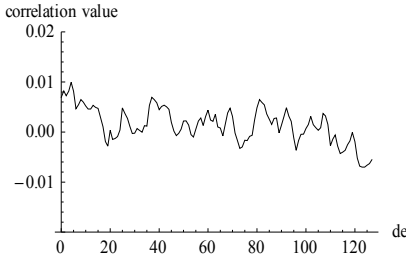
Distribuce B složky obrazu A



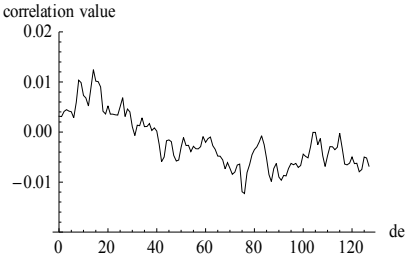
Distribuce B složky obrazu B

Entropie obrazů

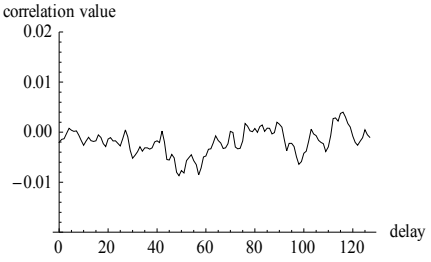
RGB	Původní obraz	Zašifrovaný obraz
R	7.56880	7.99704
G	7.28799	7.99717
B	7.82844	7.99674



Křížová korelace R složek



Křížová korelace G složek



Křížová korelace B složek

Křížová korelace sousedních pixelů pro R složku

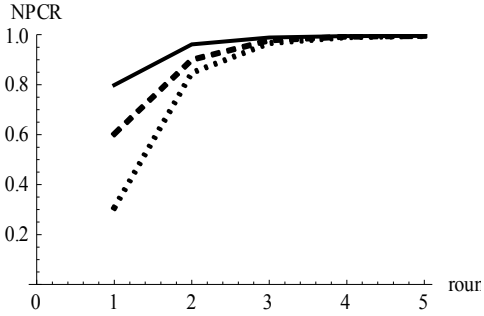
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.920450	-0.003890
Vertikální	0.932001	0.001624
Diagonální	0.872420	0.001785

Křížová korelace sousedních pixelů pro G složku

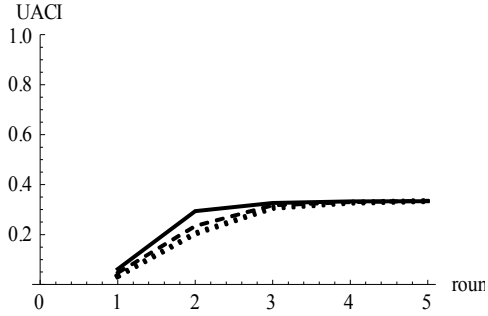
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.893017	0.002906
Vertikální	0.913575	0.009197
Diagonální	0.835212	-0.003846

Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.936993	0.000575
Vertikální	0.941166	-0.004246
Diagonální	0.893493	0.002483



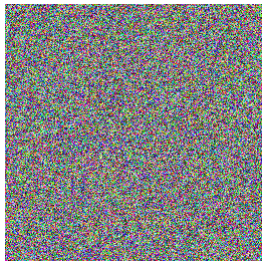
Vývoj NPCR pro šifrovací rundy



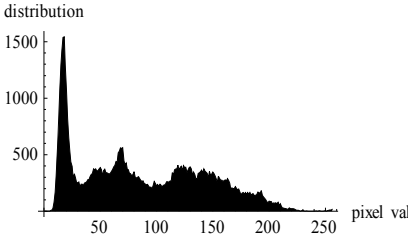
Vývoj UACI pro šifrovací rundy

10.3. Obraz „Hills“

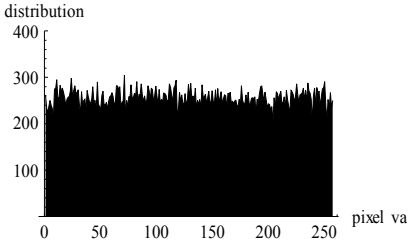
Původní obraz A



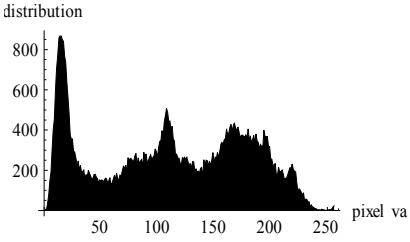
Zašifrovaný obraz B



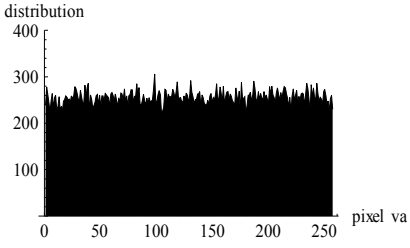
Distribuce R složky obrazu A



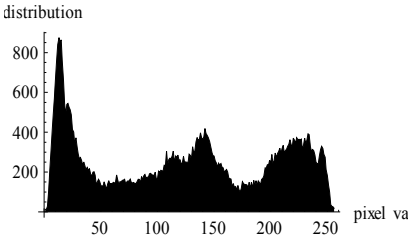
Distribuce R složky obrazu B



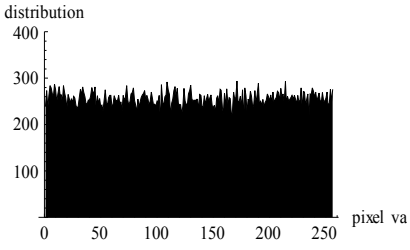
Distribuce G složky obrazu A



Distribuce G složky obrazu B



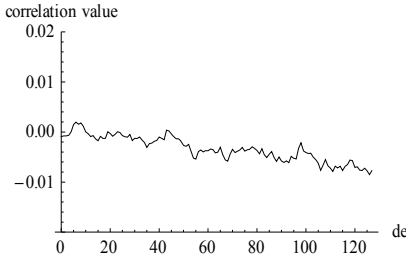
Distribuce B složky obrazu A



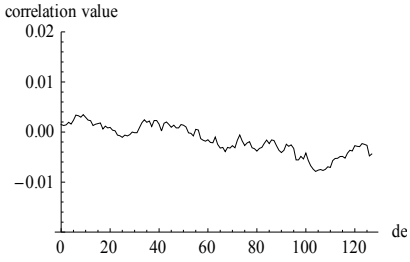
Distribuce B složky obrazu B

Entropie obrazů

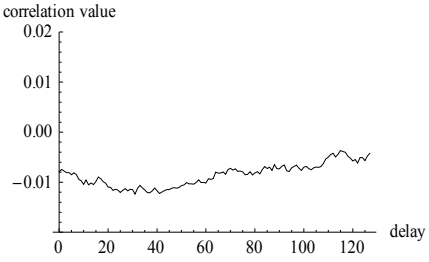
RGB	Původní obraz	Zašifrovaný obraz
R	7.44706	7.99663
G	7.70201	7.99771
B	7.80909	7.99743



Křížová korelace R složek



Křížová korelace G složek



Křížová korelace B složek

Křížová korelace sousedních pixelů pro R složku

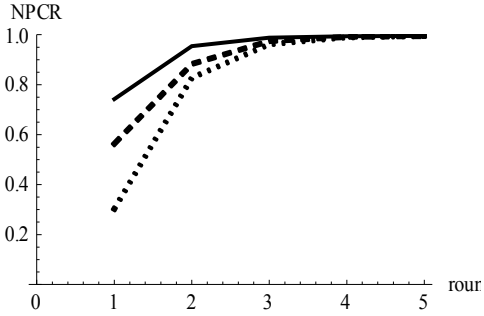
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.980822	0.001806
Vertikální	0.962296	-0.001523
Diagonální	0.963145	0.001806

Křížová korelace sousedních pixelů pro G složku

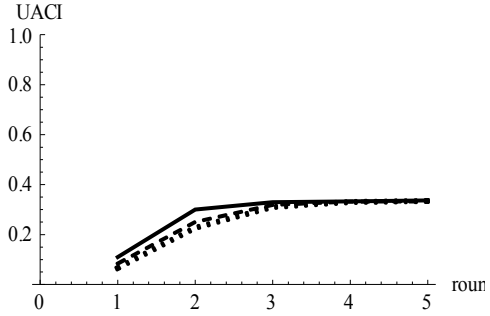
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.986186	0.008280
Vertikální	0.974575	-0.010770
Diagonální	0.975340	-0.000432

Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.992902	0.009243
Vertikální	0.988653	-0.003346
Diagonální	0.989290	0.002138



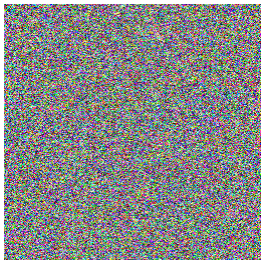
Vývoj NPCR pro šifrovací rundy



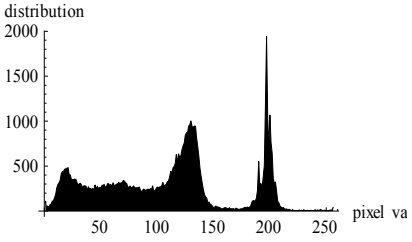
Vývoj UACI pro šifrovací rundy

10.4. Obraz „Lake“

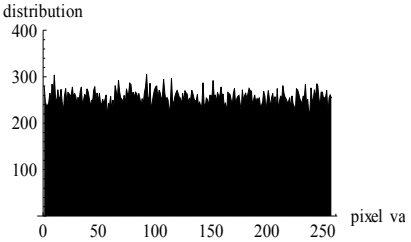
Původní obraz A



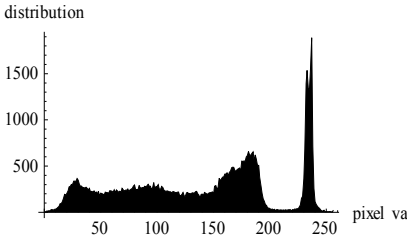
Zašifrovaný obraz B



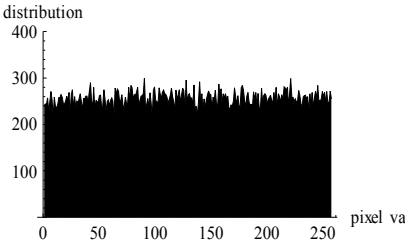
Distribuce R složky obrazu A



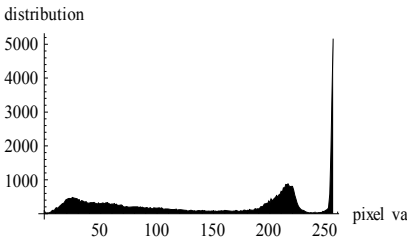
Distribuce R složky obrazu B



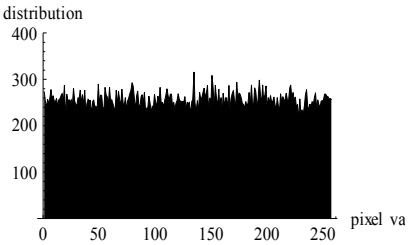
Distribuce G složky obrazu A



Distribuce G složky obrazu B



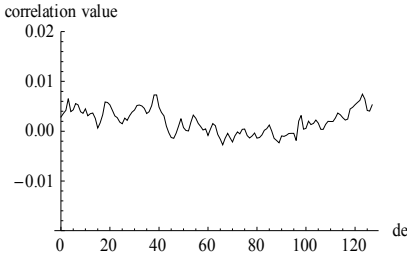
Distribuce B složky obrazu A



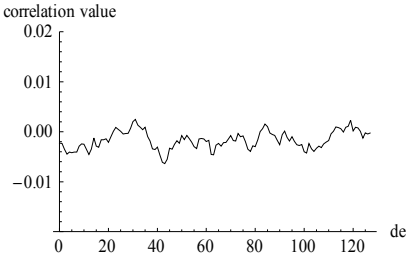
Distribuce B složky obrazu B

Entropie obrazů

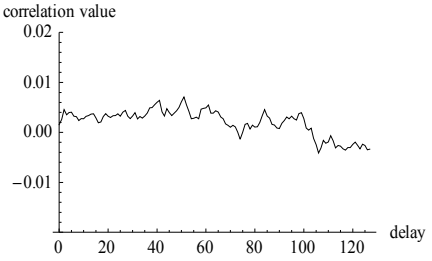
RGB	Původní obraz	Zašifrovaný obraz
R	7.29811	7.99733
G	7.47872	7.99731
B	7.21470	7.99702



Křížová korelace R složek



Křížová korelace G složek



Křížová korelace B složek

Křížová korelace sousedních pixelů pro R složku

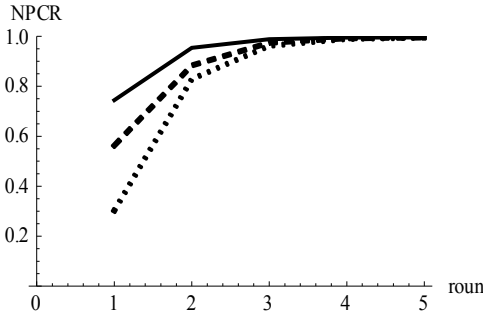
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.968294	-0.004141
Vertikální	0.956173	-0.008699
Diagonální	0.937085	-0.002314

Křížová korelace sousedních pixelů pro G složku

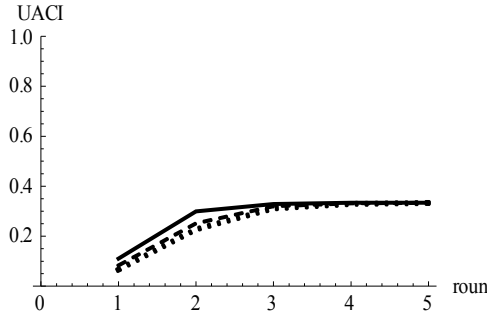
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.975394	-0.004751
Vertikální	0.965875	-0.003126
Diagonální	0.950587	-0.000511

Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.984297	0.006024
Vertikální	0.977235	0.002504
Diagonální	0.967274	0.008920



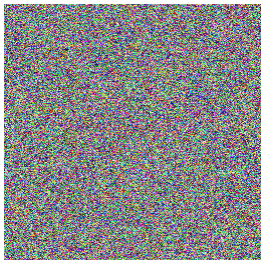
Vývoj NPCR pro šifrovací rundy



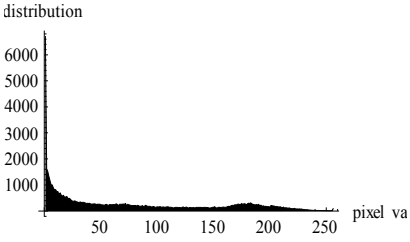
Vývoj UACI pro šifrovací rundy

10.5. Obraz „Dolphin“

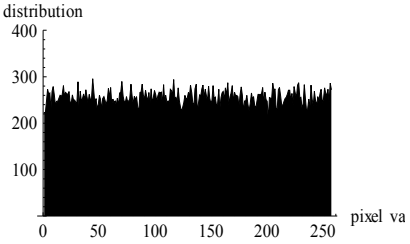
Původní obraz A



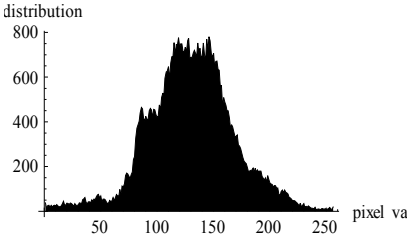
Zašifrovaný obraz B



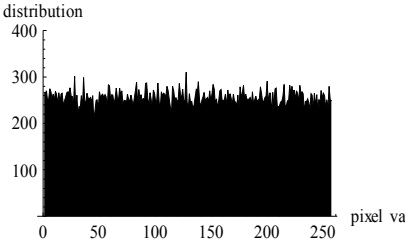
Distribuce R složky obrazu A



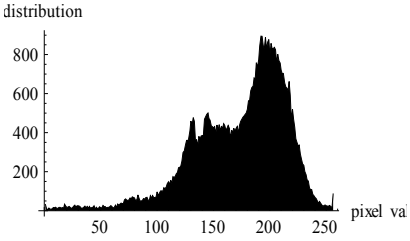
Distribuce R složky obrazu B



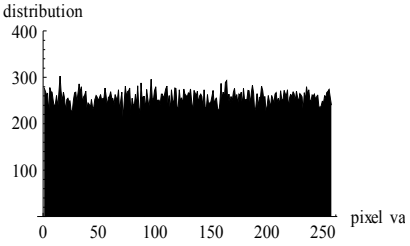
Distribuce G složky obrazu A



Distribuce G složky obrazu B



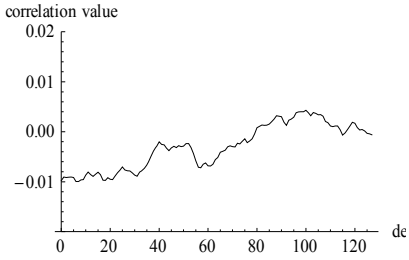
Distribuce B složky obrazu A



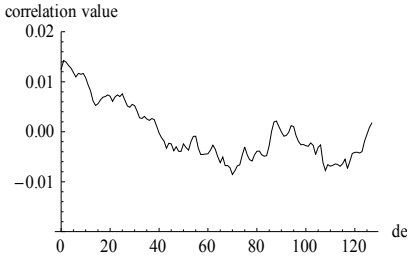
Distribuce B složky obrazu B

Entropie obrazů

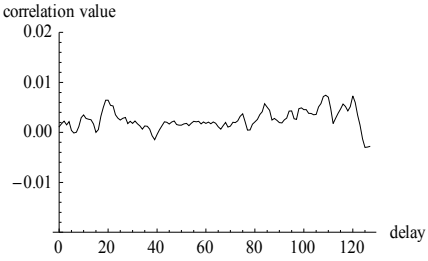
RGB	Původní obraz	Zašifrovaný obraz
R	7.25432	7.99736
G	7.25125	7.99733
B	7.22565	7.99731



Křížová korelace R složek



Křížová korelace G složek



Křížová korelace B složek

Křížová korelace sousedních pixelů pro R složku

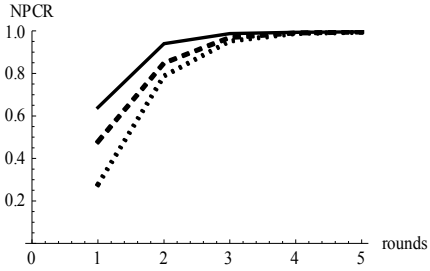
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.989950	0.000988
Vertikální	0.981156	-0.002492
Diagonální	0.970991	-0.002321

Křížová korelace sousedních pixelů pro G složku

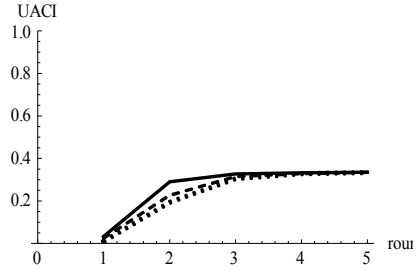
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.966877	-0.000666
Vertikální	0.941518	-0.000645
Diagonální	0.913968	-0.004129

Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.970778	-0.000854
Vertikální	0.950393	-0.013614
Diagonální	0.926812	0.000887



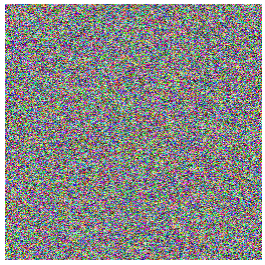
Vývoj NPCR pro šifrovací rundy



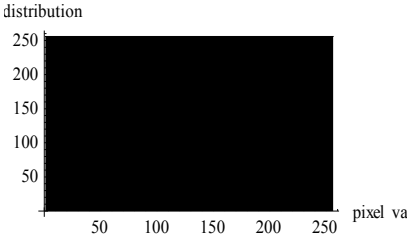
Vývoj UACI pro šifrovací rundy

10.6. Obraz „Greyscale“

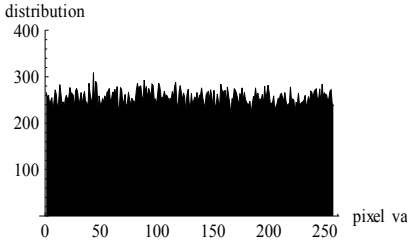
Původní obraz A



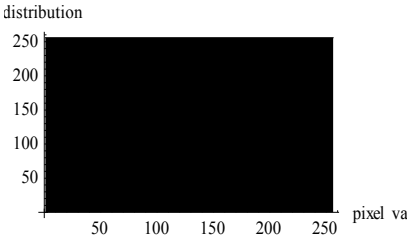
Zašifrovaný obraz B



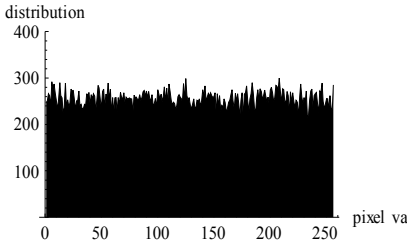
Distribuce R složky obrazu A



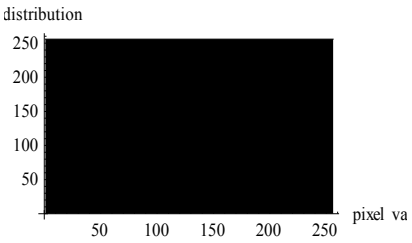
Distribuce R složky obrazu B



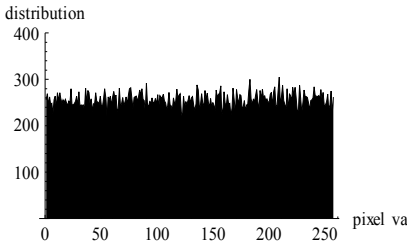
Distribuce G složky obrazu A



Distribuce G složky obrazu B



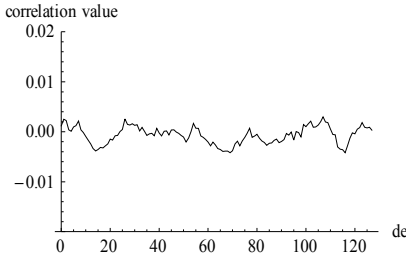
Distribuce B složky obrazu A



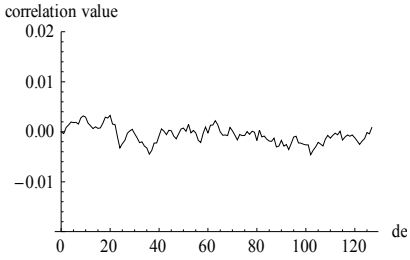
Distribuce B složky obrazu B

Entropie obrazů

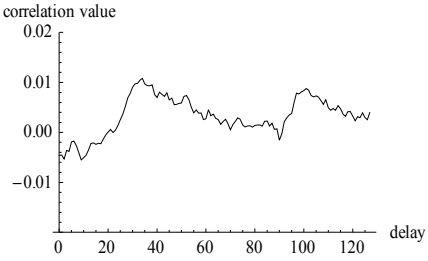
RGB	Původní obraz	Zašifrovaný obraz
R	8.0	7.99737
G	8.0	7.99692
B	8.0	7.99739



Křížová korelace R složek



Křížová korelace G složek



Křížová korelace B složek

Křížová korelace sousedních pixelů pro R složku

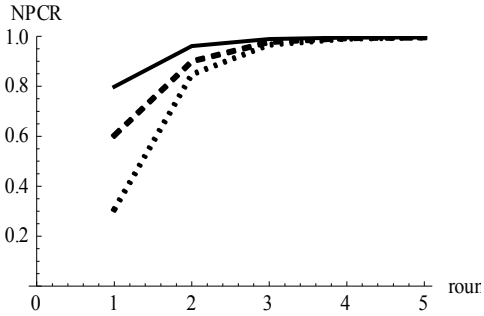
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.999999	0.000635
Vertikální	1.0	0.000437
Diagonální	0.999999	0.001362

Křížová korelace sousedních pixelů pro G složku

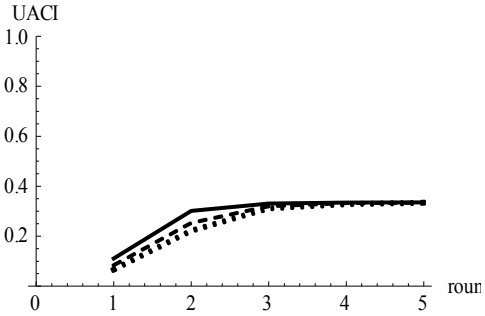
Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.999999	0.000033
Vertikální	1.0	0.000811
Diagonální	0.999999	0.000369

Křížová korelace sousedních pixelů pro B složku

Směr	Původní obraz	Zašifrovaný obraz
Horizontální	0.999999	-0.007536
Vertikální	1.0	-0.008908
Diagonální	0.999999	0.001426

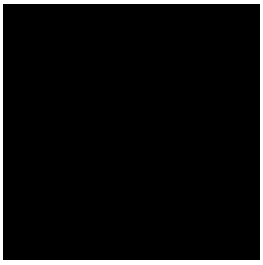


Vývoj NPCR pro šifrovací rundy

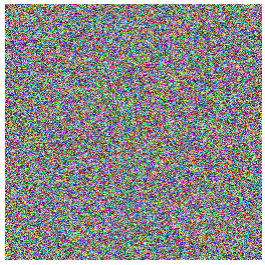


Vývoj UACI pro šifrovací rundy

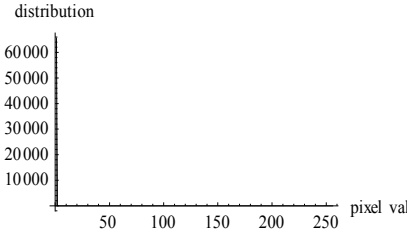
10.7. Obraz „Black color“



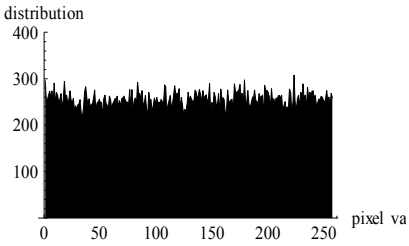
Původní obraz A



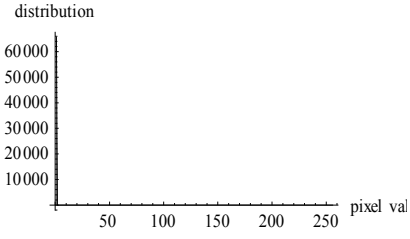
Zašifrovaný obraz B



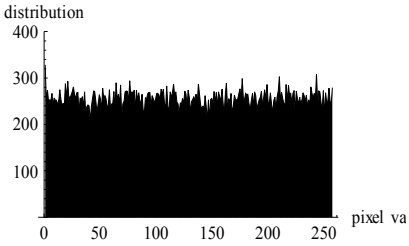
Distribuce R složky obrazu A



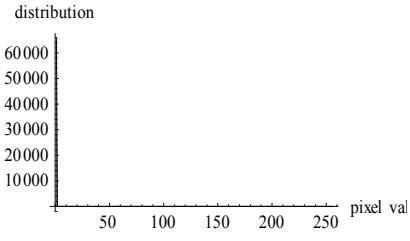
Distribuce R složky obrazu B



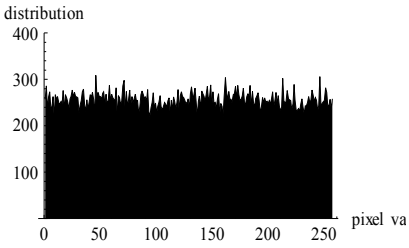
Distribuce G složky obrazu A



Distribuce G složky obrazu B



Distribuce B složky obrazu A

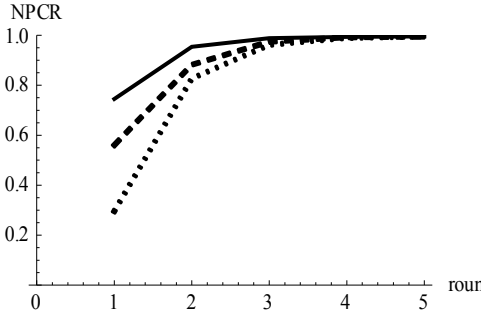


Distribuce B složky obrazu B

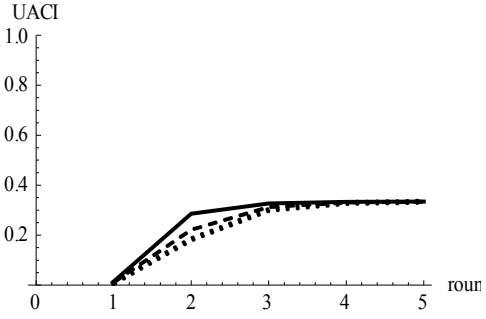
Entropie obrazů

RGB	Původní obraz	Zašifrovaný obraz
R	0.0	7.99714
G	0.0	7.99667
B	0.0	7.99701

Křížovou korelaci nelze spočítat



Vývoj NPCR pro šifrovací rundy



Vývoj UACI pro šifrovací rundy

SEZNAM AUTOROVÝCH PUBLIKAČNÍCH AKTIVIT

Články v recenzovaných časopisech

[1] GIESL, J., VLCEK, K., *Image Encryption Based on Strange Attractor*, ICGST-GVIP Journal, 2009, vol. 9, is. 2, pp. 19-26. ISSN 1687-398

[2] GIESL J., BEHAL, L., VLCEK, K., *Improving Chaos Image Encryption Speed*, SERSC-IJFGCN Journal, 2009, vol. 2, is. 3, pp. 23-36. ISSN 1738-995X

Příspěvky na konferencích

[1] GIESL, J., VLCEK, K., *Fractal Image Compression Using the Wavelet Transformation*, In IWCIT 2007, 5.-7. September 2007, Ostrava. Department of measurement and control, VŠB-TU Ostrava, 2007, ISBN 978-80-248-1567-1

[2] GIESL, J., VLCEK, K., *Audio Signal Encryption in Wavelet Domain Based on Chaotic Maps*, MENDEL 2008, 18.-20. June 2008, Brno. Faculty of Mechanical Engineering, Brno University of Technology, Brno, 2008, ISBN 978-80-214-3675-6

[3] GIESL, J., BEHAL, L., VLCEK, K., *Hardware Solution of Chaos Based Image Encryption*, Symposium on Design and Diagnostics of Electronic Circuits and Systems, 2009, pp. 198-201, ISBN 978-1-4244-3339-1

[4] GIESL, J., VLCEK, K., *Features of Fractal Image Compression*, MENDEL 2009, 24.-26. June 2009, Brno University of Technology, Brno, 2009, ISBN 978-80-214-3884-2, ISSN 1803-3814

[5] BEHAL, L., GIESL, J., *Parameters Estimation in Chaotic Synchronization By Differential Evolution Algorithm*, MENDEL 2009, 24.-26. June 2009, Brno University of Technology, Brno, 2009, ISBN 978-80-214-3884-2, ISSN 1803-3814

[6] GIESL, J., *Zrychlení algoritmu chaotické šifry pro obrazy*, PAD 2009, 9.-11. September, Soláň. Department of applied informatics, FAI Zlín, 2009, ISBN 978-80-7318-847-4

[7] PODOBA, T., GIESL, J., VLCEK, K., *Image Encryption in Wavelet Domain Based on Chaotic Maps*, CISP 2009, The 2nd International

Conference on Image and Signal Processing, TianJin University of Technology, Tianjin, China, 2009, ISBN 978-1-4244-4130-3

[8] GIESL, J., PODOBA, T., VLCEK, K., *Chaos-Based Bit Planes Image Encryption*, CISSE SCSS 2009, International Conference on Systems, Computing Sciences and Software Engineering, University of Bridgeport, USA, 2009

[9] PODOBA, T., GIESL, J., VLCEK, K., *GPU Benchmarks Based on Strange Attractors*, CISSE SCSS 2009, International Conference on Systems, Computing Sciences and Software Engineering, University of Bridgeport, USA, 2009

[10] BEHAL, L., GIESL, J., VLCEK, K., *Chaos-Based Interleaver Design*, 13th Euromicro Conference on Digital System Design, 2010, in press

[11] BEHAL, L., GIESL, J., VLCEK, K., *Cryptanalysis of Chaotic Stream Cipher by Means of Evolutional Algorithms*, MENDEL 2010, 23.-25. June 2010, Brno University of Technology, Brno, 2010, ISBN 978-80-214-4120-0, ISSN 1803-3814

ŽIVOTOPIS

OSOBNÍ INFORMACE

Jméno	Jiří Giesl
Datum narození	6.června 1983
Adresa	Mysločovice 188 763 01 Mysločovice
Stav	svobodný
Státní občanství	Česká republika
Kontakt	tel: +420 608 739 727 email: giesl.jiri@seznam.cz

VZDĚLÁNÍ

1998 – 2002	Střední průmyslová škola Zlín Obor: Management obchodu a služeb
2002 – 2005	Univerzita Tomáše Bati ve Zlíně, Fakulta technologická Obor: Informační technologie Bakalářská práce: WWW stránky pro podporu předmětu „Teorie automatického řízení I“
2005 – 2007	Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky Obor: Informační technologie Diplomová práce: Fraktální komprese statických obrazů pomocí waveletové transformace
Od 2007	Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky Obor: Inženýrská informatika, doktorské studium

JAZYKOVÉ ZNALOSTI

Český jazyk	rodilý
Anglický jazyk	aktivně
Německý jazyk	pasivně (v současné době nepoužíván)

AKADEMICKÁ ČINNOST

- Výuka magisterského předmětu "Telekomunikační systémy"
- Výuka bakalářského předmětu "Objektové programování"
- Vedoucí diplomové práce „Syntéza psychostimulačních signálů a jejich aplikace"
- Vedoucí diplomové práce "Zpracování multimediálních signálů wavelet transformací"
- Konzultant diplomové práce "Turnajový systém"
- Vedoucí bakalářské práce "Deterministický chaos v kryptografii"
- Recenzent několika příspěvků na mezinárodní konferenci CISSE SCSS 2009

ODBORNÉ ZÁJMY

- Zpracování multimediálních signálů
- Komprese, kryptografie
- Deterministický chaos a jeho aplikace
- Detekční a korekční kódy
- Evoluční algoritmy, neuronové sítě

PROGRAMÁTORSKÉ DOVEDNOSTI

- PHP, ASP.NET, XHTML, CSS, Javascript
- C/C++, C#, Basic, Pascal/Delphi, Assembler (pro řadu x86 a jiné mikroprocesory)
- WinAPI, MFC, wxWidgets, .NET (Windows Forms, základy XAML)
- OpenGL, DirectX, Allegro
- Databáze: SQL
- Základy jazyka ABAP
- ControlWeb, Mosaic, GIS
- Maple, Matlab/Simulink, Mathematica