Využití deterministického chaosu pro utajenou komunikaci - šifrování pomocí chaosu

Usage of deterministic chaos for secretive communication – encryption by means of chaos

Bc. Eva Klimková

Diplomová práce
2010

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Eva KLIMKOVÁ**
Osobní číslo:	**A08738**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Počítačové a komunikační systémy**
Téma práce:	**Využití deterministického chaosu pro utajenou komunikaci – šifrování pomocí chaosu**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na téma šifrování pomocí deterministického chaosu.**
2. **Popište systémy šifrování, co nejvíce příkladů a ukázek, včetně ukázek možností prolomení šifrovacího systému.**
3. **Vytvořte odpovídající aplikace v programu Mathematica.**
4. **Proveďte vizualizaci v programu WebMathematica.**
5. **Vytvořte prezentace obsahující popis problematiky, analýzu a grafické ukázky.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **GONZALES-MIRANDA, J. M., Synchronization And Control Of Chaos: An Introduction For Scientists And Engineers. World Scientific Publishing Company, 2004. 224 s. ISBN 978-1860944888.**
2. **HOSTE, Jim. Mathematica DeMYSTiFied. McGraw-Hill Professional, 2008. 408 s. ISBN 978-0071591447.**
3. **RUSKEEPAA, Heikki. Mathematica Navigator: Mathematics, Statistics and Graphics, Third Edition. Academic Press, 2009. 1136 s. ISBN 978-0123741646.**
4. **BENERJEE, Santo. Chaos Synchronization and Cryptography for Secure Communications: Applications for Encryption. Information Science Publishing, 2010. 350 s. ISBN 978-1615207374.**
5. **GLEICK, James. Chaos: vznik nové vědy. Ando Publishing, 1996. 350s. ISBN 80-86047-04-0.**
6. **PRIGOGINE, Ilya. Řád z chaosu: Nový dialog člověka s přírodou. Mladá fronta, 2001. 320 s. ISBN 80-204-0910-6.**
7. **HORÁK, Jiří. Deterministický chaos a jeho fyzikální aplikace. Academia, 2003. 437 s. ISBN 8020009108.**

Vedoucí diplomové práce: **Ing. Roman Šenkeřík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **19. února 2010**

Termín odevzdání diplomové práce: **7. června 2010**

Ve Zlíně dne 19. února 2010

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

L.S.

[redacted]

ABSTRAKT

Tato práce se zabývá deterministickým chaosem a jeho uplatněním v kryptografii. V první části této práce byly charakterizovány základy kryptografie. Dále byla popsána historie deterministického chaosu a jeho vlastnosti. V další části byl podrobněji rozebrán časoprostorový chaos a jeho využití pro šifrování dat. V prostředí Mathematica byl navržen nástroj pro šifrování dat pomocí časoprostorového chaosu.

Klíčová slova: deterministický chaos, časoprostorový chaos, CML, kryptografie

ABSTRACT

This work deals with deterministic chaos and its application in cryptography. In the first part of this work were characterized basic of cryptography. Further describes the history of deterministic chaos and its properties. In other parts of this work were spatiotemporal chaos analyzed and its use for data encryption. In the environment of Mathematica was designed tool to encrypt by mean of spatiotemporal chaos.

Keywords: deterministic chaos, spatiotemporal chaos, CML, cryptography

Tímto bych chtěla poděkovat Ing. Romanu Šenkeříkovi Ph.D. za poskytnuté praktické i teoretická rady a za odbornou pomoc při zpracování této diplomové práce. Dále bych ráda poděkovala své rodině za jejich podporu při studiu.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně …….

Podpis diplomanta

OBSAH

ÚVOD

Pokud nahlédneme do výkladového slovníku, dozvíme se, že pojem chaos označuje zmatek, nepořádek, stav s vysokou neuspořádaností a nepředvídatelností. Pojem determinismus označuje vlastnost procesu, jehož každý stav je určen předcházejícím. V této práci bude tedy podrobněji popsán deterministický chaos, tedy složitý systém, který je předvídatelný a každý jeho stav je vymezen stavem předcházejícím.

Teorie chaosu, je poměrně nová vědní disciplína, která se začala prosazovat teprve v polovině 20. století, nicméně její kořeny sahají až do 19. století.

Další vědní disciplínou, která bude popsána, je časoprostorový chaos. Je to chaotický systém, který šíří chaotické chování jak v prostoru, tak v čase.

Úkolem této práce bylo popsat základní metody kryptografie. Objasnit problematiku deterministického chaosu a popsat jeho vlastnosti a chování. Dále bylo potřeba charakterizovat časoprostorový chaos a jeho vlastnosti. Popsat jeho využití pro utajenou komunikaci a realizovat praktickou ukázku kryptosystému s využitím časoprostorového chaosu.

I. TEORETICKÁ ČÁST

1 ÚVOD DO KRYPTOGRAFIE

Potřebu šifrovat svoje zprávy mělo lidstvo už několik let před naším letopočtem. V minulosti to byli nejčastěji důvody vojenské či politické. V dnešní době internetu tato potřeba mnohonásobně stoupá. Velká spousta lidí posílá po internetu velmi důvěrná data, například data o svém bankovním účtu, a proto potřeba zabezpečení informací se prudce zvýšila.

1.1 Co je to kryptografie

Kryptografie je tedy věda zabývající se převáděním zpráv do nečitelné podoby. Vedle ní je další obor kryptoanalýza, který se pokouší dešifrovat zprávu bez znalosti klíče. Oba tyto obory slučuje kryptologie.

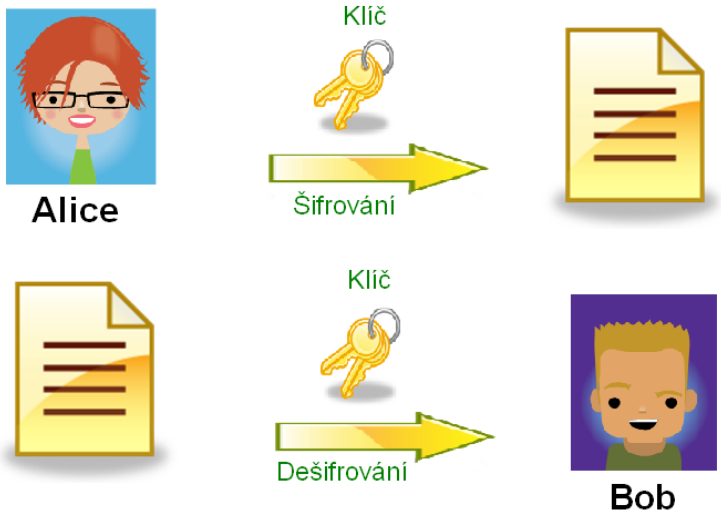
Moderní kryptografii velmi pozdvihl Claude Elwood Shannon, který v polovině 20. století stanovil základy pro šifrování. Vyslovil větu: síla algoritmu spočívá na pilířích matematické složitosti a ne na tajnostech kolem něj [5]. V 70. letech byla kryptologie uznána za vědní obor a byla teoreticky navržena asymetrická kryptografie. Prakticky byla provedena až o několik let později.

Šifrování je tedy postup, při kterém převádíme text do nečitelné podoby. K tomu zpravidla využíváme tzv. klíč, s kterým danou zprávu zakódujeme a následně i dekódujeme. Bez znalosti tohoto klíče by mělo být prakticky nemožné zprávu dekódovat.

1.2 Symetrická kryptografie

Symetrická kryptografie používá pro šifrování i dešifrování stejný klíč (Obr. 1). Tato metoda je výpočetně jednodušší než asymetrická kryptografie a proto i rychlejší. Má však i své nevýhody. Odesilatel a příjemce se musí předem dohodnout na tajném klíči. Při delších vzdálenostech nastává problém s distribucí klíče.

Tato metoda se dnes používá především pro zálohování dat. Malé firmy a domácnosti mohou používat klíč dlouhý 40 bitů. Pro větší firmy a data s vysokým utajením se doporučuje klíč 64 bitů, běžně se však používá klíč 128 bitů a více. [10]



Obrázek 1 Symetrická kryptografie

1.3 Asymetrická kryptografie

Při asymetrickém šifrování se používají dva různé klíče, jeden veřejný pro šifrování a druhý soukromý pro dešifrování. Tyto dva klíče musí být odlišné a při znalosti jednoho klíče, nesmí být možné odvodit klíč druhý.

Šifrování pak probíhá následovně, Bob uveřejní svůj veřejný klíč. Alice použije tento veřejný klíč pro zašifrování zprávy. Bob po přijetí zprávy vezme svůj soukromý klíč a dešifruje zprávu (Obr 2).

Pokud známe veřejný klíč, nesmí být možné z něj odvodit klíč soukromý ani jinak dešifrovat tajnou zprávu.

Tato metoda je výpočetně náročnější a pomalejší, má však i své velké výhody. Odpadá nám problém se sdílením jediného klíče.

Asymetrická šifra byla poprvé použita v 70. letech 20. století. Nejznámější je šifra RSA, která využívá toho, že rozložit velké číslo na prvočísla je velmi náročné, zato jejich součin je velmi snadný. Při použití dostatečně velkého klíče je šifra stále bezpečná [10].

1.4 Blokové šifry

Jedním ze způsobů jak urychlit šifrování, je šifrovat data po blocích. Blokové šifry pracují s pevně stanovenou délkou dat, nejčastěji 128 bitů, kterou naráz zašifrují. Symetrické šifry, které pracují po blocích, jsou například DES, AES, IDEA, Blowfish a Twofish.

Obrázek 2 Asymetrická kryptografie

1.5 Proudové šifry

Proudové šifry oproti blokovým šifrují každý bit samostatně. Nejčastěji se tento způsob používá u asymetrických šifer. Nicméně mezi algoritmy proudových a blokových šifer může být jen malý rozdíl. Některé algoritmy se mohou za určitých podmínek chovat jako blokové i jako proudové.

1.6 Eliptické křivky

V roce 1985 byla představena nová metoda využívající veřejný klíč. Eliptické křivky na rozdíl od asymetrické kryptografie nevyužívají modulární aritmetiku ale operace nad eliptickou křivkou. Tato metoda je daleko bezpečnější a efektivnější, při srovnatelné bezpečnosti potřebuje daleko kratší klíč než předchozí metody [10].

1.7 Kvantová kryptografie

Další nová metoda šifrování je kvantová kryptografie. Ta jde úplně jinou cestou, neřeší problém pomocí matematických metod, ale využívá přírodních zákonů. Snaží se o bezpečnou distribuci klíčů mezi odesilatelem a příjemcem.

Každý skutečný komunikační kanál má určitou fyzickou realizaci, svou fyzikální podstatu. Odposlouchávání kanálu z hlediska fyziky odpovídá procesu měření určitých veličin. Z kvantové mechaniky jako jeden ze zásadních důsledků vyplývá, že jakékoliv měření systém ovlivňuje, mění jeho stav. Tuto změnu je možné fyzikálními metodami zjistit, což

znamená, že v určitých situacích je možné spolehlivě detekovat odposlech. Dnes jsou známé a vyzkoušené dvě metody, jak toho docílit. Jedna je založena na měření polarizace fotonů, druhá na zvláštních vlastnostech stavu propletenosti [15].

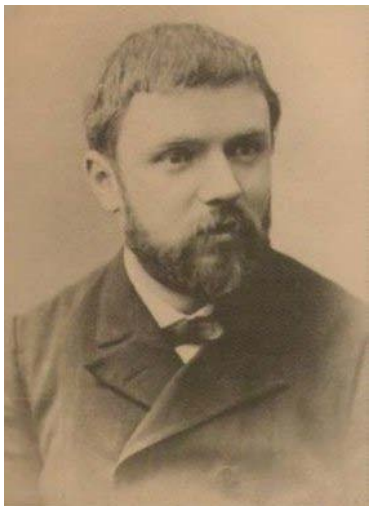
2 HISTORIE CHAOSU

Na počátku 20. století se zrodila nová věda, teorie chaosu. Jako první pozoroval chaotický pohyb francouzský vědec Henri Poincaré. Po něm se studiem dynamických systémů zabývalo několik dalších vědců. Trvalo však delší dobu, než vědci přestali svět zkoumat jen pomocí lineárních systémů, které mají ustálený průběh. Stejně tak trvalo mnoho let, než vědci přestali náš svět chápat jen pomocí Euklidovské geometrie. Ta dokáže definovat přímky, kružnice či obdélníky, ale v přírodě nenajdeme takovéto pravidelné obrazce. Pouze pomocí fraktální geometrie můžeme napodobit přírodu. [5]

2.1 Henri Poincaré

Francouzský matematik, fyzik, astronom a filozof Henri Poincaré (1854 - 1912) jako první pochopil možnost existence chaosu. Věnoval se studiu dynamických systémů a topologie. Spolu s Albertem Einsteinem položili základy speciální teorie relativity.

Henri Poincaré se narodil 29. 4. 1854 v Nancy. Vystudoval Polytechnickou a důlní univerzitu. Krátký čas působil jako profesor matematiky na univerzitě v Caen. Po zbytek svého života působil na univerzitě v Sorbonně [16].



Obrázek 3 Henri Poincaré

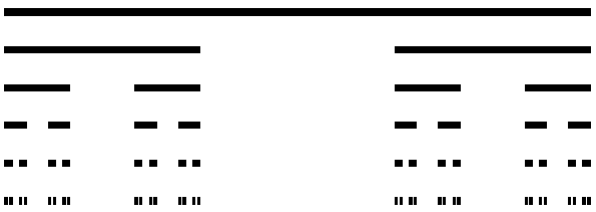
Při studiu problému tří těles došel k závěru, že tento problém je často neřešitelný. Problém dvou těles popsal už Newton a dokonale ho vyřešil. Můžeme si představit například Slunce a Zemi. Země se pohybuje kolem Slunce v dokonalé elipse a lze její polohu kdykoli

spočítat. Pokud do systému přidáme třetí těleso, to začne svou gravitací působit na předchozí tělesa a problém se tak mnohonásobně zvětší. Pohyby tří těles se dají spočítat a sledovat velmi dlouho, až do doby než převládne neurčitost. Henri Poincaré konstatoval, že chování tří těles z dlouhodobého hlediska nelze předpovědět [5].

2.2 Georg Cantor

Německý matematik Georg Cantor se především proslavil svojí prací v teorii množin. Zajímalo ho nekonečno a nekonečně velké množiny. V teorii množin zavedl pojmy ordinérní a kardinální číslo.

Objevil takzvané Cantorovo diskontinuum, neboli Cantorovu množinu (Obr. 4). Vezmeme úsečku od 0 do 1, odstraníme z ní prostřední třetinu, pak odstraníme prostřední třetinu ze zbylých dvou segmentů a tak budeme pokračovat až do nekonečna. Získáme tak Cantorovu množinu, která má vlastnosti fraktálu. Mnoho věcí v přírodě se chová jako Cantorova množina, například chyby v přenosu mohou mít takovéto rozdělení [5].



Obrázek 4 Cantorova množina

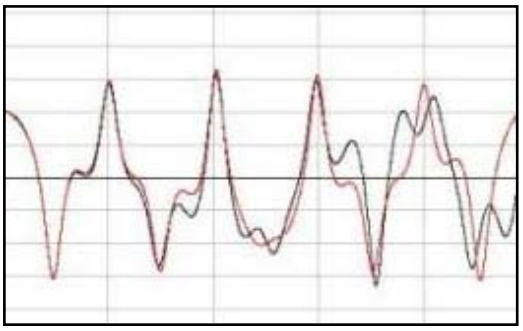
2.3 Edward Lorenz

Edward Norton Lorenz (1917 - 2008) byl americký matematik a meteorolog. Zabýval se teorií chaosu, objevil podivný atraktor a je spojován s termínem motýlí efekt.

Edward Lorenz se narodil 23. května 1917 ve státě Connecticut. Studoval matematiku na střední škole v New Hampshiru a později na Harvardské univerzitě v Candgridge. Za druhé světové války působil jako meteorolog. Tento obor se mu zalíbil a rozhodl se mu nadále věnovat [17].

V roce 1960 začal Lorenc studovat počasí na svém novém sálovém počítači. Počítač nebyl zdaleka tak výkonný, aby mohl dokonale modelovat počasí na Zemi, ale i při zjednodušených podmínkách se začalo simulované počasí velmi podobat tomu reálnému.

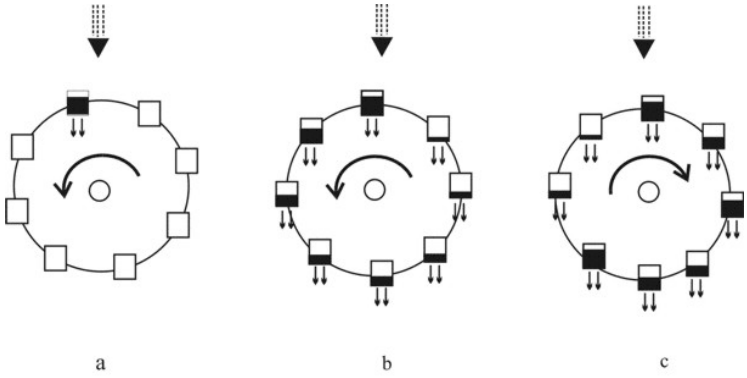
Jednoho dne se Lorenz rozhodl zkoumat detailněji jednu část grafu, proto zadal do počítače počáteční podmínky, které už jednou počítač zpracovával. K jeho velkému překvapení obdržel graf, který se podobal prvnímu jen svým začátkem a ke konci se naprosto rozcházel s původním. Rozdíl byl způsoben zadáním hodnot s pouhými třemi desetinnými místy, zatím co počítač počítal s šesti místy. Tato malá počáteční odchylka měla za následek naprosto rozdílné počasí. Lorenz pochopil, že dlouhodobě předpovídat počasí bude naprosto nemožné. Takto vznikl Motýlí efekt: něco tak malého jako mávnutí motýlím křídlem může způsobit tajfun na druhé straně zeměkoule.



Obrázek 5 Motýlí efekt

Dalším chaotickým systémem, který Lorenz zkoumal, bylo mechanické vodní kolo. Toto jednoduché zařízení má nečekaně složité chování (Obr. 6). Vodní kolo je vybaveno nádobami s propustným dnem. Pokud je proud vody tekoucí shora pomalý, voda vyteče z nádoby a kolo se vůbec neroztočí. Je-li proudění rychlejší, horní naplněná nádoba uvede kolo do pohybu a to se může otáčet konstantní rychlostí. Je li proudění ještě rychlejší, kolo se může začít otáčet v opačném směru a systém se stává chaotickým.

Lorenz zjistil, že z dlouhodobého hlediska se smysl otáčení může změnit mnohokrát, otáčení se nikdy neustálí v rovnoměrném tempu a nikdy se neopakuje předpověděným způsobem. Grafické zobrazení tvoří tzv. Lorenzův podivný atraktor, který představuje jakousi nekonečnou složitost (Obr. 15) [5].



Obrázek 6 Mechanické vodní kolo

2.4 Mitchell Feigenbaum

Mitchell Jay Feigenbaum (1944) je matematik a fyzik který se věnoval chaotickým zobrazením a zavedl Feigenbaumovu konstantu.

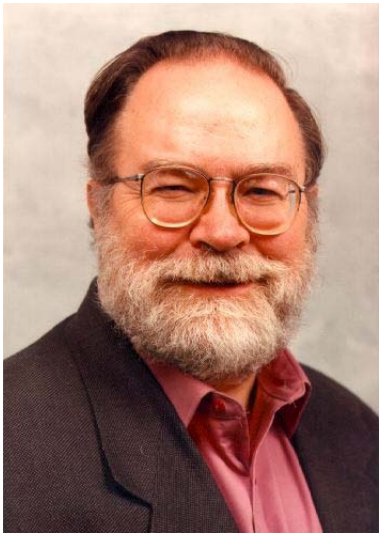
Feigenbaum se narodil ve Philadelphii jako syn polským a ukrajinským rodičům. Navštěvoval střední školu v Brooklynu v New Yorku a později vysokou školu v New Yorku. V roce 1970 získal doktorát na Massachusettském technickém institutu. Po krátké činnosti na Cornell univerzitě ve Virginii dostal práci v Los Alamoském výzkumném institutu, který se věnoval výzkumu turbulence [18].

Obrázek 7 Mitchell Feigenbaum

Mnoho matematických zobrazení obsahujících jednoduchý lineární parametr začne vykazovat pro hodnoty parametru z určité oblasti zcela náhodné chování známé jako chaos. Jakmile se parametr blíží k této oblasti, zobrazení vykazuje bifurkace pro přesné hodnoty parametru. Nejprve existuje jeden stabilní bod, pak pro určitou hodnotu parametru začne řešení oscilovat mezi dvěma body, pro další hodnotu parametru dojde k nové bifurkaci a řešení začne oscilovat mezi čtyřmi body atd. V roce 1975 Feigenbaum s použitím svého počítače HP-65 objevil, že poměr rozdílu mezi dvěma následnými hodnotami parametru, ve kterých dochází k bifurkaci, se postupně blíží ke konstantě přibližně 4.6692. Feigenbaum poskytl matematický důkaz tohoto faktu a ukázal, že stejné chování a stejná konstanta se objevuje v široké třídě matematických funkcí vedoucích k chaosu. Byl to jeden z prvních kroků vedoucích k popisu nepředvídatelného chaosu. Tento „poměr konvergence“ je nyní znám jako Feigenbaumova konstanta [5].

2.5 James Yorke

James A. Yorke je univerzitní profesor, matematik a fyzik na Marylendské univerzitě. Narodil se v roce 1941 v Plainfield, New Jersey. Navštěvoval univerzitu v Hillside, New Jersey. Říká se, že Yorke objevil Lorenze a dal této vědě její jméno, chaos.

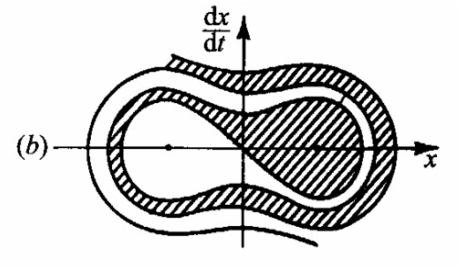


Obrázek 8 James Yorke

Věnoval se mimo jiné rozhraním fraktálních pánví. Například Lorenzův model má jen jeden atraktor, jeden druh chování, který převládne po přechodu systému do rovnovážného

stavu a přitom je to chaotické chování. Některé odlišné systémy mohou skončit u nechaotického rovnovážného chování ale s více rovnovážnými stavy. Yorke se snažil zjistit, v kterém rovnovážném stavu systémy skončí.

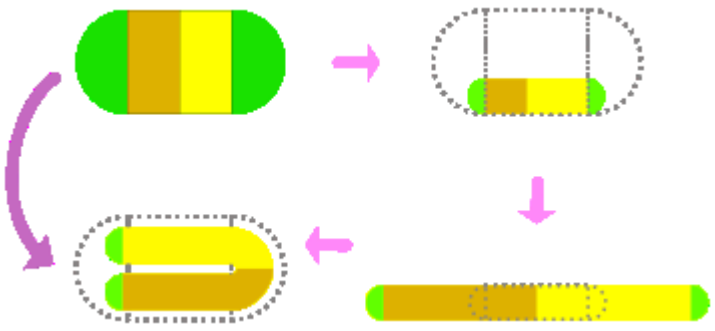
Představme si hrací automat s mechanickou pákou. Automat měl nakloněnou rovinu, několik gumových překážek a elektrické nárazníky, které dodávají míčku energii. Podle toho jak natáhneme páku, se vystřelí míček a skončí v jedné ze dvou děr. Pokud natáhneme páku minimálně, míček skončí v levé díře, pokud bude páka natáhnutá maximálně, skončí míček v pravé díře. Yorke se snažil sestavit závislosti počáteční polohy páky a konečného stavu. Pokud míček spadl do levé díry, zakreslil bílý bod, pokud spadl do pravé, zakreslil černý bod. Dostaneme fraktální obrazec, který má nekonečné množství detailů. Některé části obrazce budou jen černé nebo bílé. Pokud si zvětšíme některé části, objevíme další černé body v bílé oblasti a naopak [5].



Obrázek 9 Rozhraní fraktálních pánví

2.6 Stephen Smale

Smale se narodil 15. července 1930 v městě Flint, v Michiganu. Přes třicet let působil jako profesor matematiky na univerzitě v Californii.



Obrázek 10 Smaleovy podkovy

Smale začal svou kariéru na univerzitě v Chicagu, kde se věnoval studiu topologie. Později se také zabývat dynamickými systémy. Zkoumal trajektorie systémů ve fázovém prostoru a pozoroval, jak se její části natahují a zkracují. Výsledkem jeho výzkumu bylo vynalezení Smaleovy podkovy. Jedná se o prostor, který v jednom směru natáhneme a v druhém stlačíme a pak jej složíme. Opakováním procesu vzniká strukturované mísení známé všem, kdo někdy dělali lístkové těsto. Dva body, které se nakonec ocitnou u sebe, mohli být původně velmi vzdálené. Tato topologická transformace vytvořila základ pro pochopení chaotických vlastností dynamických systémů. Avšak fyzikům se tato transformace zdála příliš umělá a trvalo velmi dlouho, než ji přijali [5].

3 DETERMINISTICKÝ CHAOS

Většina lidí si pod pojmem chaos představí neuspořádaný a nepředvídatelný děj. Avšak přírodní systémy, které můžeme označit pojmem deterministický chaos, se chovají podle přesně daných pravidel, která jsou často velmi složité. Příklad deterministického chaosu lze nalézt v přírodě (chování atmosféry, turbulence tekutin a mravenčí kolonie), v ekonomii (Elliotovy vlny) nebo třeba v chemii (chemické reakce).

3.1 Co je to chaos

Teorie chaosu se tedy zabývá chováním nelineárních dynamických systémů, které za jistých podmínek vykazují jev známý jako deterministický chaos. Charakteristickou vlastností chaosu je citlivost na počáteční podmínky. V důsledku této citlivosti se chování těchto fyzikálních systémů, jeví jako náhodné, i když model systému je deterministický v tom smyslu, že je dobře definovaný a neobsahuje žádné náhodné parametry [14].

3.2 Hemiltoniánské versus disipativní systémy

Jako první se začali studovat Hemiltoniánská systémy, tedy systémy které neztrácejí energii. To musíme ale brát s určitou rezervou, protože všechny systémy časem ztrácejí energii. Pouze pokud budeme zkoumat systém v krátkém časovém úseku, můžeme disipaci energie zanedbat a nazvat ho Hemiltoniánský.

Typickým představitelem takovýchto systémů je naše sluneční soustava, která z hlediska ztráty energie je pro nás konzervativní. Teorii chaosu na těchto systémech studoval například Poincaré nebo Boltzman. Velkou výhodou Hemiltoniánské systémů je jejich menší výpočetní náročnost, zatím co disipativní systémy se zřídkakdy obejdou bez výpočetní techniky.

My se budeme zabývat především disipativními systémy, tedy takovými, které s časem ztrácí energii. Tyto dynamické systémy jsou pak popsány soustavami diferenciálních rovnic, které umožňují syntézu jejich řízení. Vlastní výraz, deterministický chaos, plyne z faktu, že systémy které jej produkují, lze modelovat a tím, že se v těchto modelech nevyskytuje náhodnost. Typickým příkladem je například Lorenzův atraktor, který je generován třemi diferenciálními rovnicemi (1). Struktura těchto rovnic je přesně známa, a i přesto tato sestava jednoduchých diferenciálních rovnic generuje chaotické chování bez ohledu na počáteční podmínky [11].

$$\frac{dx}{dt} = \sigma(y - x),$$

$$\frac{dy}{dt} = -xz + rx - y, \tag{1}$$

$$\frac{dz}{dt} = xy - bz.$$

3.3 Chaotický pohyb

Předpokladem pro chaotický pohyb systému jsou tři základní vlastnosti

- citlivost na počáteční podmínky
- topologická tranzitivita
- husté periodické orbitaly

3.3.1 Citlivost na počáteční podmínky

I velmi malá změna počátečních podmínek u chaotického pohybu vede po delším čase k naprosto odlišným výsledkům (viz. Edward Lorenz 2.3). Pouze pro naprosto stejné počáteční podmínky dostaneme stejné výsledky. Tento jev se také nazývá motýlí efekt. Mávnutí motýlích křídel nad Tokiem může způsobit uragán nad New Yorkem.

3.3.2 Topologicky tranzitivní systém

Tranzitivita znamená, že aplikace transformace na libovolný daný interval I_1 ho roztahuje až do doby, kdy překryje libovolný další daný interval I_2.

Tranzitivita, husté periodické body a citlivost na počáteční podmínky se dají rozšířit na libovolný metrický prostor. J. Banks a jeho kolegové ukázali v roce 1992, že v nastavení obecného metrického prostoru tranzitivita a zároveň husté periodické body implikují citlivost na počáteční podmínky.

Tento elementární, ale neočekávaný fakt vedl Bau-Sen Du, z Matematického institutu, Academia Sinica, v Taiwanu k definici silnější verze citlivé závislosti - k extrémně citlivé závislosti - která není důsledkem tranzitivity a hustých periodických bodů. Extrémně citlivá závislost znamená, hrubě řečeno, že blízké body se oddělují a konvergují nekonečně často, což je mnohdy právě případ chaotických dynamických systémů [14].

3.4 Vizualizace chaotického systému

3.4.1 Atraktor

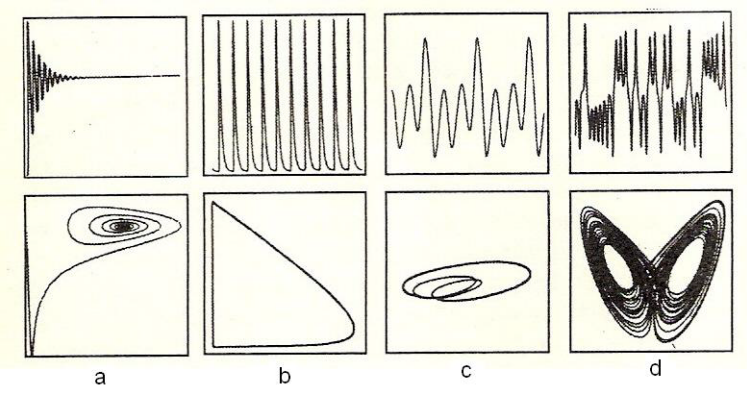
Pohyb libovolného dynamického systému lze popsat jako trajektorii ve fázovém prostoru. Po určitém čase vykreslí systém křivku, pokud je spojitý, nebo množinu bodů, pokud je sytém diskrétní. Atraktor je tedy křivka fázového prostoru neboli konečný stav systému.

Trajektorie se může vyvíjet několika způsoby, může směřovat do jednoho bodu, například mechanické kyvadlo, na které působí tření a časem se zastaví. Tento případ nazýváme množina bodů a je to nejjednodušší případ atraktoru.

Těleso se také může ustálit tak, že osciluje mezi několika stavy. Tomuto případu říkáme periodické body (spočitatelné), nebo kvaziperiodické body (nespočitatelné). Může jít například o cizí těleso v naší sluneční soustavě, které je přitakováno sluncem, začne obíhat kolem slunce a časem se jeho dráha ustálí.

Dalším případem jsou chaotické atraktory, ty jsou obtížně dopředu předvídatelné, jelikož jsou velice citlivé na počáteční podmínky. Chaotické však neznamená náhodné, jde o deterministické systémy, které jsou určeny svými počátečními podmínkami, ale jsou velmi složité.

Chaotický pohyb může vést také na podivný atraktor, který je velmi složitý a vykazuje velké detaily. Příkladem může být mechanické vodní kolo, které zkoumal Edward Lorenz (viz 2.3). Je to jeden z nejsložitějších atraktorů, který vykresluje obrazec podobný motýlím křídlům.

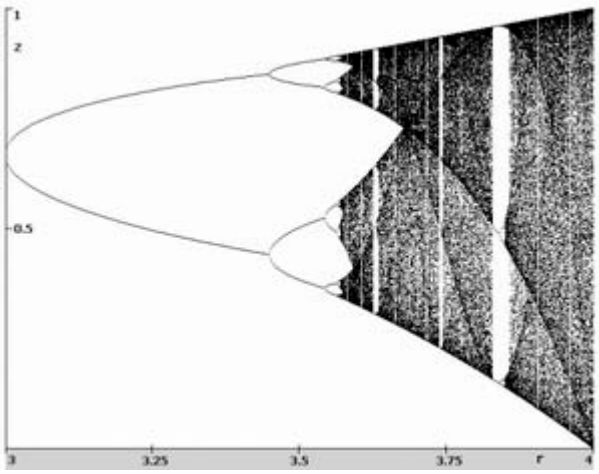


Obrázek 11 Příklady atraktorů, a) množina bodů, b) c) periodické body, d) podivný atraktor

3.4.2 Bifurkační diagram

Znázornit chaotické chování můžeme také pomocí bifurkačního diagramu. Bifurkace vyznačuje jev, při kterém dochází k velkým změnám vnitřního stavu ve sledovaném systému v případě, že se vstupní parametry nepatrně změní. U některých systémů se při změně vstupních parametrů systém mění jen minimálně nebo dokonce lineárně. V takových případech k bifurkacím nedochází.

U jiných systémů po dosažení kritických hodnot na vstupu, dochází k prudké změně chování systému. Příkladem může být vznik turbulence při proudění kapalin, po dosažení kritických hodnot.

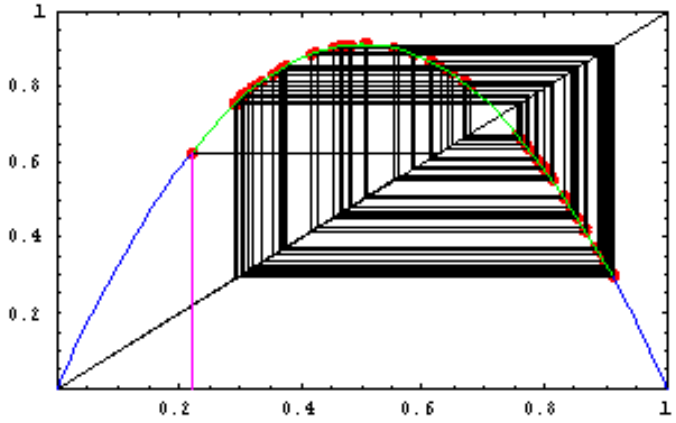


Obrázek 12 Bifurkační diagram

3.4.3 WEB diagram

Jeden ze způsobů jak znázornit chaotické chování je také WEB diagram. V tomto grafu představuje přímka jdoucí z počátku množinu tzv. pevných bodů, které mají charakter odpuzování či přitahování. Pokud by takový např. přitahující bod byl dosazen do logistické rovnice, pak by funkční hodnota byla stejná jako argument [11].

Vlastní křivka znázorňuje samotnou logistickou rovnici s tím, že směrnice jejího průniku s přímkou (množinou pevných bodů) udává charakter příslušného bodu, ale také i chování systému - rovnice v jeho okolí [11].



Obrázek 13 WEB diagram

3.5 Chaotické systémy

3.5.1 Logická rovnice

Jedním z nejjednodušších modelů chaotického chování je logická rovnice (2). Tato rovnice vznikla při výzkumu uzavřených biologických systémů. Parametr *r* zde představuje plodnost populace, tendenci populace růst ale i klesat.

$$x_{n+1} = rx_n(1 - x_n) \tag{2}$$

Jako příklad si můžeme představit rybník s kapry. Pokud v rybníce vysadíme několik jedinců kaprů, bude jejich počet v několika prvních letech prudce stoupat, protože kapři budou mít dostatek potravy. V dalších letech se nárůst populace zpomalí, protože kapři už nemají tak ideální podmínky. Po dosažení kritické hranice bude populace střídavě klesat a stoupat, dokud nedosáhne rovnovážného stavu.

Tato rovnice se nemusí vždy ustálit na jediné hodnotě, v závislosti na parametru *r* může oscilovat, nebo dojít do chaotického chování.

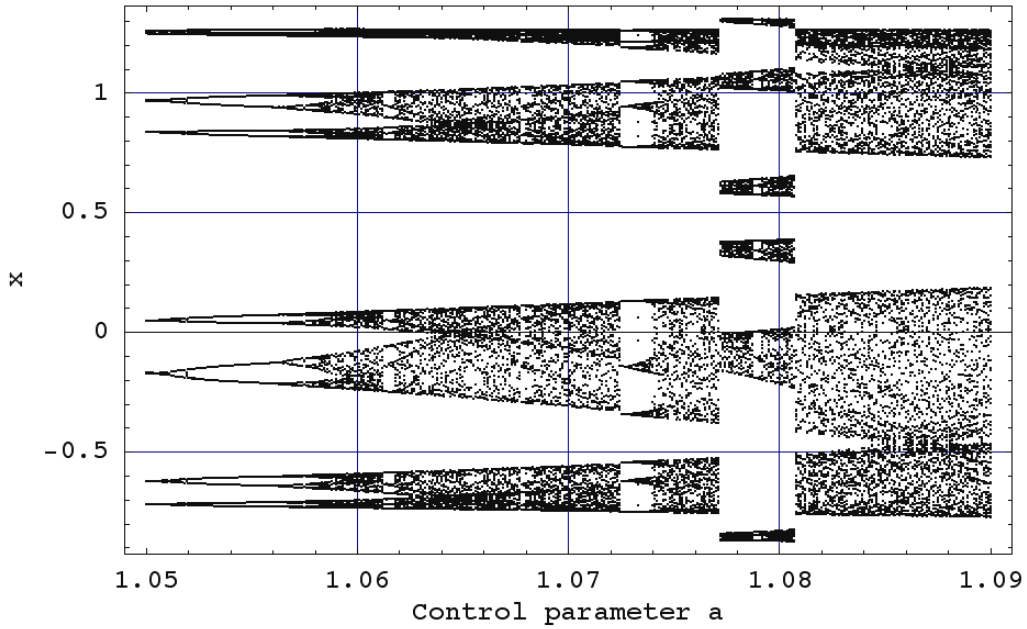
3.5.2 Henonova mapa

Henonova rovnice (3) je diskrétní dynamický systém, který byl představen jako zjednodušený model Poincarého map pro Lorenzův systém. Je to jeden z nejvíce studovaných dynamických systémů, které vykazují chaotické chování. Jedná se o dvou dimenzionální rozšíření jedno rozměrné kvadratické mapy.

$$x_{n+1} = 1 + y_n - ax_n^2$$

$$y_{n+1} = bx_n \tag{3}$$

Mapa závisí na dvou parametrech, *a* a *b*, který pro standardní Henonovu mapu mají hodnoty *a=1,4* a *b =0,3*. Pro tyto hodnoty Hénonova mapa vykazuje chaotický pohyb (Obr. 14). Pro jiné hodnoty parametrů může být mapa chaotická, oscilující, nebo konvergují k ustálené dráze.



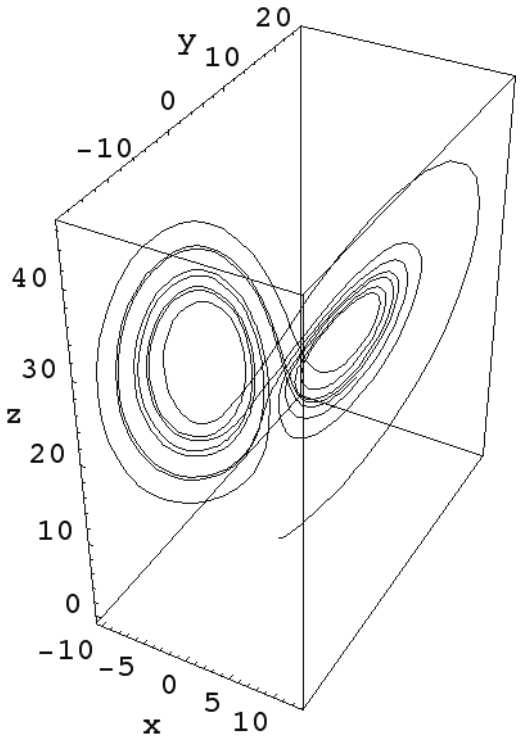
Obrázek 14 Bifurkační diagram Henonovy mapy

3.5.3 Lorenzův systém

Lorenzův systém (4) je trojdimenzionální deterministický dynamický systém, odvozený ze zjednodušení rovnice vynucené konvence v atmosféře. Byl zaveden Edward Lorenz v roce 1963, který ji použil ke studiu předpovědi počasí.

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{4}$$

Tento systém pro určité parametry vykazuje chaotické chování a znázorňuje Lorenzův podivný atraktor (Obr. 15). Parametr σ se nazývá Prandtlovo číslo a parametr ρ se nazývá Rayleighovo číslo.



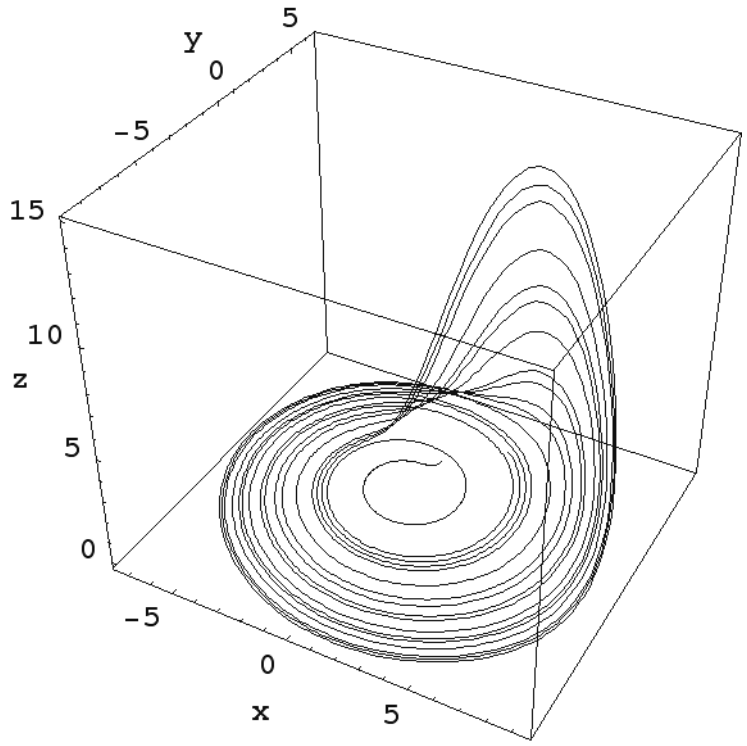
Obrázek 15 Lorenzův atraktor

3.5.4 Rösslerův systém

Rösslerův systém (5) je systém tří nelineárních diferenciálních rovnic. Tyto diferenciální rovnice definují spojitý dynamický systém, který vykazuje chaotické chování. Chová se podobně jako Lorenzův atraktor, ale umožňuje jednodušší kvalitativní analýzu a má pouze jeden orbital. Atraktor byl navržen v roce 1976, teoretické rovnice byly později použity pro modelování rovnováhy v chemických reakcích.

$$\begin{aligned} \frac{dx}{dt} &= -y - z \\ \frac{dy}{dt} &= x + ay \\ \frac{dz}{dt} &= b + z(x - c) \end{aligned} \tag{5}$$

Rössler studoval chaotický atraktor s parametry $a=0,2$, $b=0,2$ a $c=5,7$, který zobrazuje jednoduchý chaotický atraktor s trajektorií rotující kolem pevného bodu (Obr. 16).



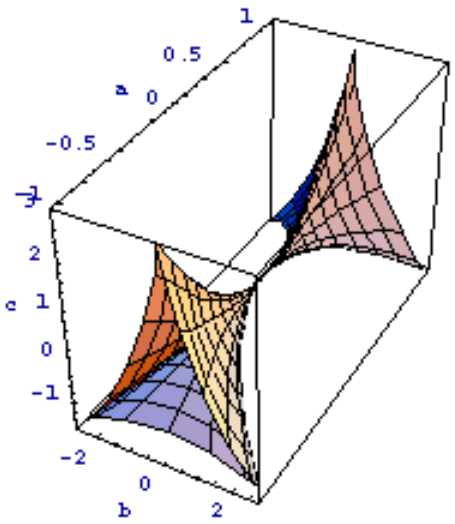
Obrázek 16 Rösslerův atraktor

3.6 Teorie katastrof

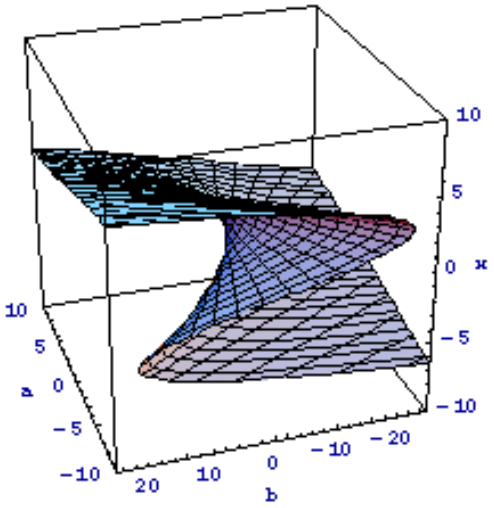
V šedesátých letech 20. století byla vytvořena teorie katastrof francouzským matematikem Rene Thomem. Tato teorie pozoruje dynamické systémy a zkoumá jejich prudké změny v závislosti na plynulých změnách řídících parametrů. Teorie katastrof zkoumá reálné systémy a snaží se určit tzv. katastrofickou množinu, která určuje, za jakých podmínek dojde ke katastrofě. Pomocí této teorie můžeme vysvětlit jak z ryze deterministického systému, při souvislých a malých změnách systému, může dojít přes bifurkace do stavu chaosu.

Pomocí katastrofy záhyb (Obr. 18) si můžeme lehce vysvětlit princip katastrof. Každý bod této plochy je určen třemi parametry *a, b* a *x*. Parametry *a* a *b* mohou představovat například teplotu, tlak, apod. Parametr *x* bude závisle proměnná např. stav systému, nebo mechanická pevnost. Jestliže se mění hodnoty parametrů, např. *a (-10 do +10)* a *b (-20 do +20)*, pak stav daného systému lze chápat jako bod, který se pohybuje po ploše v závislosti na parametrech. Při hodnotách $b > 20$ a zároveň $a < 0$ se stane podivná věc. Pro tuto a větší hodnotu je *x* definováno mnohem výše, než tomu bylo doposud, což znamená, že systém přejde skokem z jednoho stavu do druhého. Právě tento děj nazýváme katastrofa.

U složitějších systémů, které mají více řídicích parametrů je interpretace podobná, jako například u katastrofy typu pyramida (Obr. 17). Plocha tohoto tvaru je tvořena množinou bodů, které představují katastrofu. Jinými slovy, pokud se v daném dynamickém systému mění tři parametry, pak tzv. parametrická trajektorie v tomto parametrickém prostoru dynamického systému při průchodu plochou indikuje vznik katastrofy, která může znamenat např. právě změnu charakteru singulárního bodu systému neboli bifurkaci [11].



Obrázek 17 Katastrofa typu pyramida



Obrázek 18 Katastrofa typu záhyb

4 ČASOPROSTOROVÝ CHAOS

V poslední době časoprostorový chaos (Spatiotemporal chaos) přitahuje stále větší pozornost. Příklady praktického využití časoprostorového chaosu jsou například umělé neuronové sítě, hydrodynamika tekutin, plazmové systémy, chemické reakce a šíření populace biologických druhů.

CML (coupled map lattices) je časoprostorový chaotický systém, který bude dále podrobněji rozebrán. Časoprostorový chaotický systém je prostorově rozšířený systém, který může vykazovat chaos jak v prostoru, tak v čase. Ten je často modelován pomocí parciálních diferenciálních rovnic, obyčejných diferenciálních rovnic, nebo CML.

CML je považován za základní model časoprostorového chaotického systému. CML je dynamický systém, který je diskrétní v čase a prostor. Je složen z nelineárních map umístěných na mřížkách, kterým se říká lokální mapy. Každá lokální mapa je spojena s další lokální mapou, za určitých podmínek. Vzhledem k vnitřní nelineární dynamice jednotlivých lokálních map a šíření prostorových spojů mezi jednotlivými mapami, může CML vykazovat časoprostorový chaos. Použití CML jako modelu časoprostorového chaotického systému má dvě hlavní výhody. První je, že CML zachycuje nejdůležitější rysy časoprostorového chaosu, další je, že CML lze snadno ovládat jak analyticky tak numericky [8].

Jeden z nejpopulárnějších CML systémů je definován podle rovnice (6)[8].

$$x_{n+1}^{j} = (1-\varepsilon) f\left(x_n^i\right) + \frac{\varepsilon}{2}\left[f\left(x_n^{j+1}\right) + f\left(x_n^{j-1}\right)\right] \tag{6}$$

Kde funkce f reprezentuje rovnici s chaotickým chováním, zde bude použita logická rovnici (2). Parametr r logické rovnice je roven 4. Proměnná x^j_n reprezentuje prvek j-té úseku $(j = 1, 2, \ldots, L)$, kde L je počet dílů v CML mřížce. Proměnná n reprezentuje čas $(n = 0,1,2, \ldots)$, a proměnná $\varepsilon \in (0,1)$, znázorňuje intenzitu sil řídících chaos. Struktura CML je zobrazena na obrázku (Obr. 23).

5 SYNCHRONIZACE CHAOSU

Synchronizace chaotických systémů se stala velmi populárním oborem, vzhledem k jeho využití v komunikaci. Se synchronizací chaotických systémů se můžeme setkat také v jiných oborech, například v biologii (synchronizace kardio-respiračního systému).

Synchronizace chaosu je jev, který může nastat, pokud dva, nebo více chaotických oscilátorů jsou vzájemně provázané, nebo když jeden chaotický oscilátor působí na jiný chaotický oscilátor. Vzhledem k motýlímu efektu, který způsobuje odlišnost trajektorií dvou identických chaotických systémů, které začali s téměř stejnou počáteční podmínkou, se může zdát chaotický systém vyvíjející se v synchrony dosti překvapivý. Nicméně, synchronizace dvou signálů nebo řízený chaotický oscilátor je jev zavedený v praxi a dobře teoreticky popsán. Dále je popsáno několik hlavních typů synchronizace.

5.1 Úplná synchronizace

Úplná synchronizace (Identical / Complete Synchronization CS) je první objevenou a nejjednodušší formou synchronizace. Úplná synchronizace nastává, pokud se shodují stavové proměnné všech propojených systémů bez časového posunutí. Tedy pokud dva systémy $X=\{x_1, x_2, x_3, \ldots x_n\}$ a $Y=\{y_1, y_2, y_3, \ldots y_n\}$, kde x_i a y_i jsou stavové proměnné, jsou totožné $X \equiv Y$, nebo pokud synchronizační chyba definovaná jako: $|X(t) - Y(t)| \rightarrow 0$ asymptoticky.

5.2 Lineární synchronizace

Lineární synchronizace (Linear Generalized Synchronization LGS) se používá především u dvou strukturálně odlišných chaotických systémů. Lineární synchronizace nastane pokud $X=\Phi Y$, kde Φ je lineární transformační funkce, která převádí stavové proměnné systému Y na sytém X.

5.3 Fázová synchronizace

Fázová synchronizace (Phase, Imperfect Phase Synchronization PS) ovlivňuje pouze fázi synchronizovaných systémů, zatím co amplituda zůstává nekorelovaná. Je to nejčastější případ synchronizace vyskytující se v přírodě. Fázová synchronizace u neperiodických chaotických systémů je značně náročná, protože je obtížné určit jejich fázi. Pokud však

atraktor obou systémů je rotující kolem jednoho bodu, dá se fáze atraktoru určit podle rovnice (7).

$$\varphi(t) = arctan(y(t)/x(t)) \tag{7}$$

5.4 Zpětnovazební synchronizace

Zpětnovazební synchronizace (Lag Synchronization LS) nastává, pokud je jeden systém korelován s druhým se zpožděním τ, tedy $Y(t) = X(t + \tau)$.

Synchronizace s časovým posunem, nemusí být jen se zpožděním, ale i obrácená (anticipated synchronization). V tomto případě přijímač předvídá činnost vysílače. Takováto synchronizace má daleko zajímavější využití než synchronizace se zpožděním.

6 CHAOS A JEHO POUŽITÍ V KRYPTOGRAFII

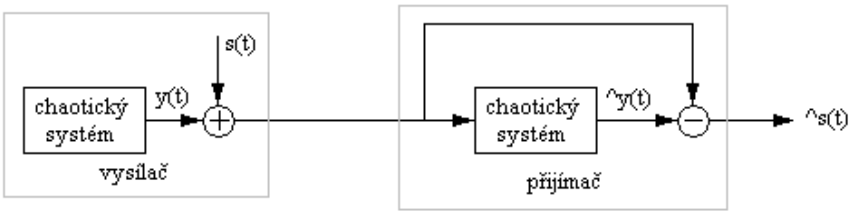
Deterministický chaos má pro kryptografii několik velmi vhodných vlastností. Především se využívá jeho citlivosti na počáteční podmínky. I při nepatrné změně vstupních dat, za použití stejné mapovací funkce, dostaneme rozdílné výsledky. Vzhledem k tomu, že jako klíč používáme reálná čísla, je počet všech možných řešení opravdu enormní. Přesnost reálných čísel také závisí na použitém tytu hardware. Systém tedy bude odolný proti útoku hrubou silou.

6.1 Chaotická modulace

Modulace signálu je proces, při kterém se signál převede na takový, který je vhodný pro přenos v daném prostředí. Při modulování chaotického signálu se využívají dva způsoby. Modulátorem může být samotný chaotický signál, který je ovlivňován informačním signálem. V druhém případě se smíchá chaotický a informační signál na nízké frekvenci a pak se klasickým modulátorem převede na vysokou frekvenci.

6.1.1 Chaotické maskování

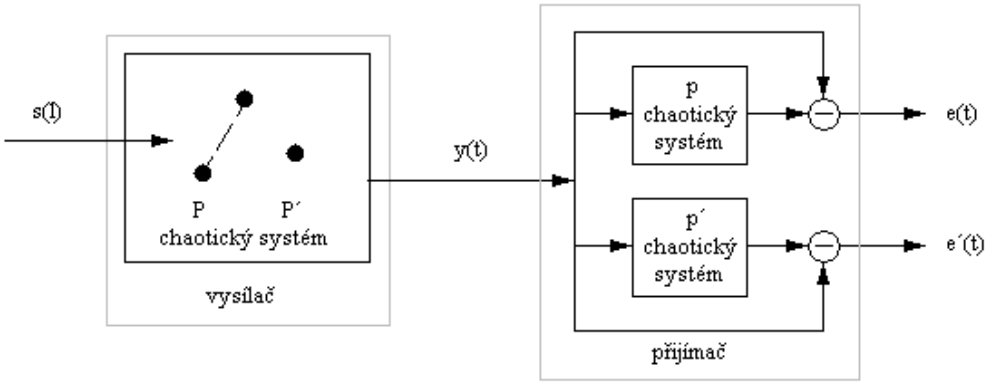
Chaotické maskování (chaotic masking) se provádí pomocí smíchání informačního a chaotického signálu. Jeho obnovení se děje pomocí synchronizace. Vysílač a přijímač je vybaven totožnými chaotickými systémy a synchronizace je dosaženo tak, že přijímač je řízen pomocí přijímaného signálu. Pomocí synchronizace se vytváří signál *^y(t),* který je podobný chaotickému signálu *y(t).* Informační signál *s(y)* proto můžeme získat odečtením výstupního signálu synchronizace od přijímaného signálu. Avšak tento systém pracuje spolehlivě jen tehdy, pokud je informační signál vůči chaotickému zanedbatelně malý, tak že ho upraví jen málo. Ovšem kvůli šumu v kanále je obtížné obnovit signál, takže se tato metoda příliš nepoužívá.



Obrázek 19 Chaotické maskování

6.1.2 Chaotické klíčování

Při chaotickém klíčování (chaotic shift keying, CSK) moduluje informační signál některý parametr chaotického signálu. Například informační signál může být binární a bude nabývat dvou hodnot *p* a *p´*. Bit bude znázorněn chaotickou křivkou určité délky. Bit posléze dekódujeme tak, že budeme přicházejícím signálem synchronizovat oba systémy s vektorem p a p´. Pokud se jeden systém synchronizuje, detekuje tak přijímaný signál.



Obrázek 20 Chaotické klíčování

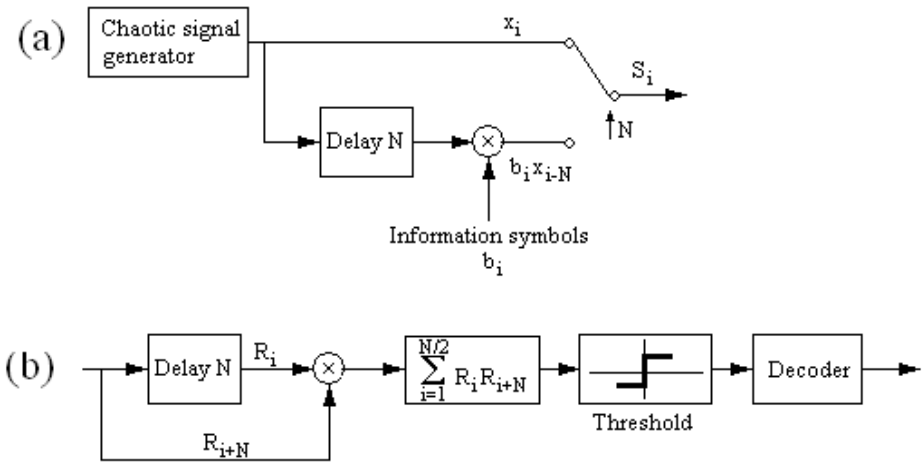
6.1.3 Diferenční chaotické klíčování

V roce 1996 G. Kolumban navrhl jednu z nejjednodušších modulačních technik diferenční chaotické klíčování (differential chaos shift keying, DCSK). Tato metoda využívá chaotický signál jako nosnou vlnu pro přenos digitální informace. Při vyslání jednoho bytu se chaotickým systémem vygeneruje signál určité délky a ten je vysílán po přenosovém kanálu. Pokud budeme vysílat znak 1, bude za ním vyslána jako kopie. Pokud bude přenášena 0, bude za ní přenesena její kopie sečtena s -1. Přijímač obě části signálu porovná a vyhodnotí.

Signál, který přenášíme po kanálu lze vyjádřit rovnicí (3), kde b_1 je bitová hodnota, která může nabývat hodnoty ±*1*.

$$S_i = \begin{matrix} x_i \\ b_1 x_{i-n} \end{matrix} \qquad \begin{matrix} 0 < i \leq N \\ N < i \leq 2N \end{matrix} \tag{3}$$

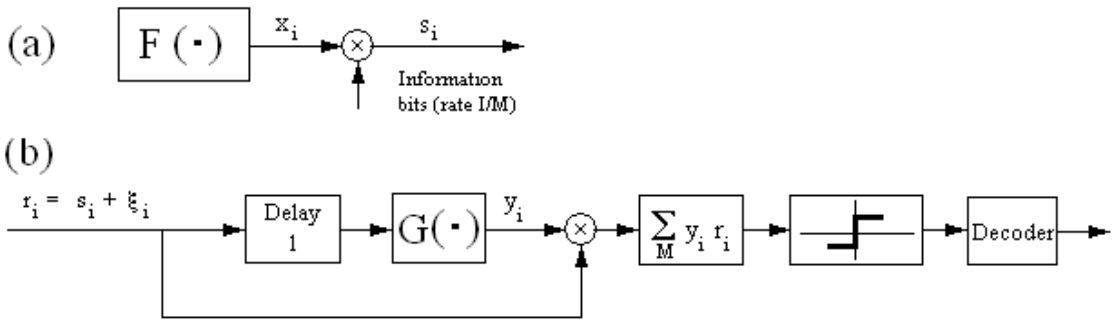
Nevýhodou této metody je, že polovinu času vysíláme referenční křivku. Dále mohou nastat problémy při realizaci, protože musíme použít zpožďovací členy a spínače. I přes to je to nejvýhodnější metoda synchronizace.



Obrázek 21 DCSK a) modulace, b) demodulace

6.1.4 Symetrické chaotické klíčování

Další metoda využívá modulaci signálu pro synchronizaci chaosu. Využívá dvou chaotických signálů, u kterých je jedna podmínka, a to aby transformace mapy byla sudá funkce, tedy *F(x) = F(-x).*



Obrázek 22 schéma SCSK a) modulace b) demodulace

SCSK má velice dobré vlastnosti v závislosti na síle signálu. Ale je nutné upravit chaotické systémy tak, aby jejich součin byl sudou funkcí. Dále systém jako takový musí být stabilní. Přesto se vlastnosti SCSK jeví jako velmi dobré.

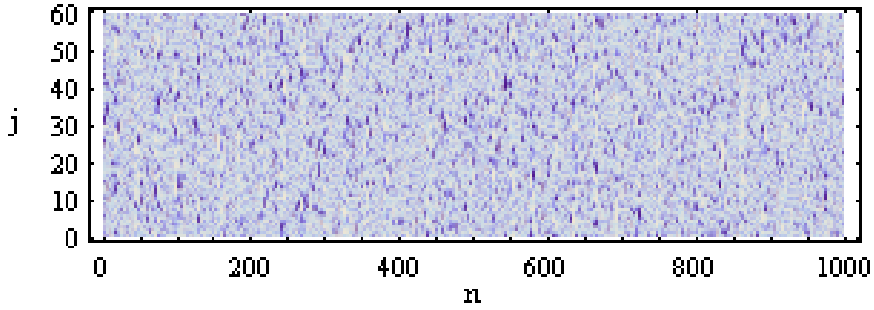
6.2 Využití CML systémů v kryptografii

Časoprostorové chaotické systémy mohou být úspěšně použity pro konstrukci kryptosystémů. Při použití typických časoprostorových chaotický systém, tj. CML můžeme dosáhnout uspokojivých vlastností. Toto téma bude podrobněji popsáno v následující kapitole 7.

II. PRAKTICKÁ ČÁST

7 CML SYSTÉMY A JEJICH VYUŽITÍ V KRYPTOGRAFII

V posledních desetiletích byly často zkoumány vlastnosti chaosu a jeho využití v kryptografii. Hlavní výhodou těchto systémů je citlivost na počáteční podmínky. V této kapitole se budeme zabývat časoprostorovým chaosem a jeho využití v kryptografii. Velmi vhodné pro tyto účely jsou CML systémy, které jsou popsány v kapitole 4. Nejpoužívanější CML systém můžeme popsat rovnicí (6). Na obrázku (Obr. 23) můžeme vidět grafické znázornění CML systému, kde bílý bod reprezentuje hodnotu 0 a tmavě modrý hodnotu 1.



Obrázek 23 Znázornění CML

7.1 Šifrování užitečného signálu pomocí CML

Pro šifrování užitečného signálu bude využit OCOML (one-way coupled open map lattice) (Obr. 24) popsaný rovnicí (8) [9].

$$x_{n+1}^{j} = (1-\varepsilon) f\left(x_n^j, a_j\right) + \varepsilon f\left(x_n^{j-1}, a_{j-1}\right) \tag{8}$$

$$x_n^1 = a_j$$

Kde funkce f představuje chaotický systém, v tomto případě bude použita logická rovnice (2), kde parametr $r = 4$, $\varepsilon = 0{,}95$ a $L = 60$.

Proměnná a_j, kterou je možné charakterizovat jako vektor $a = (a_1, a_2, \ldots, a_L)$, bude získána z rovnice OCRML (one-way coupled ring map lattice) (9) (Obr. 25) [9].

$$z_{n+1}^{j} = (1-\varepsilon) f\left(z_n^j\right) + \varepsilon f\left(z_n^{j-1}\right) \tag{9}$$

$$z_n^{j+L} = z_n^j$$

Chaotický signál bude digitalizován pomocí rovnice (10).

$$S_n^j = int\left[x_n^j \times 2^u\right] mod 2^v \tag{10}$$

Kde $u,v \in N$. Šifrovaný kód G vznikne vynásobením chaotického signálu S a užitečného signálu M (10).

$$G_n^j = M_n^j S_n^j \tag{11}$$

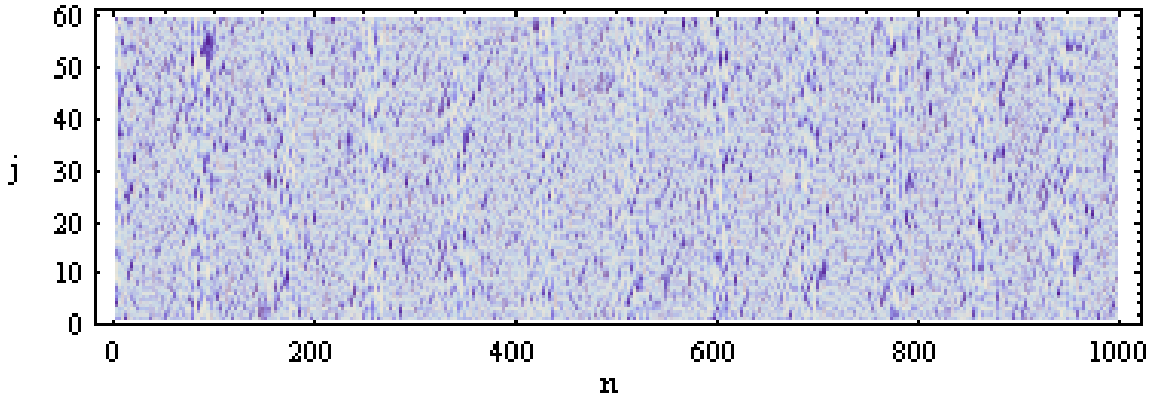
Dešifrování signálu bude probíhat velmi podobně, nejprve musí být získán chaotický signál S, který slouží jako klíč. Po vynásobení klíčového signálu a šifrovaného signálu, vznikne opět původní signál.

$$y_{n+1}^j = (1-\varepsilon) f\left(y_n^j, a'_j\right) + \varepsilon f\left(y_n^{j-1} a'_{j-1}\right)$$

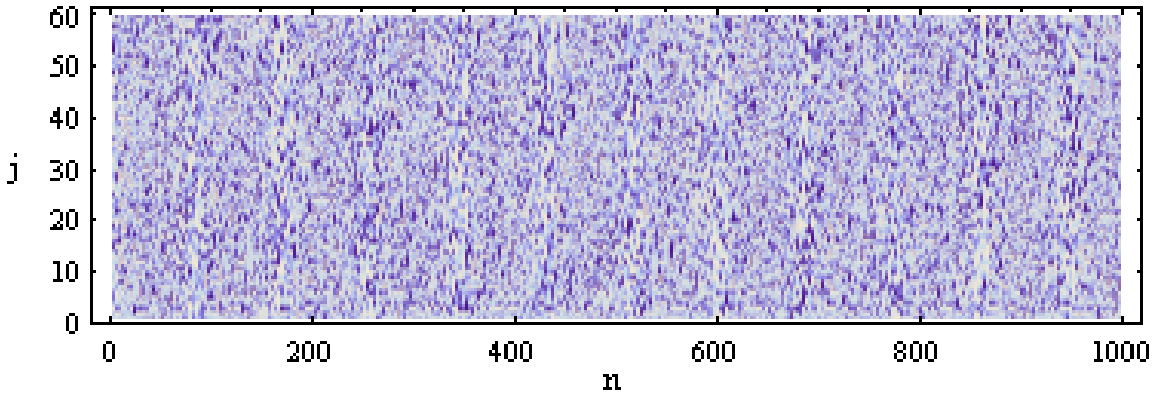
$$S'^j_n = int\left[y_n^j \times 2^u\right] mod 2^v \tag{12}$$

$$M'^j_n = G_n^j S_n^j$$

Pokud $a' = a$, $y_n^j = x_n^j$, pak jsou tyto dva CML systémy synchronizované, to znamená že y_n^j a x_n^j produkují stejný chaotický signál $S'^j_n = S_n^j$. V důsledku toho můžeme získat opět otevřený text $M'^j_n = M_n^j$ [8].



Obrázek 24 Struktura OCRML



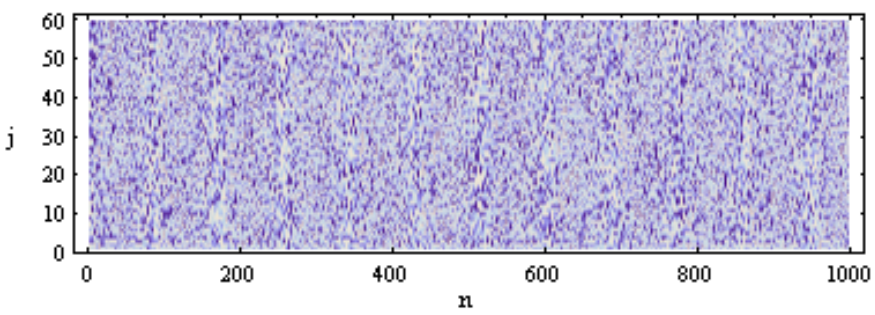
Obrázek 25 Struktura OCOML

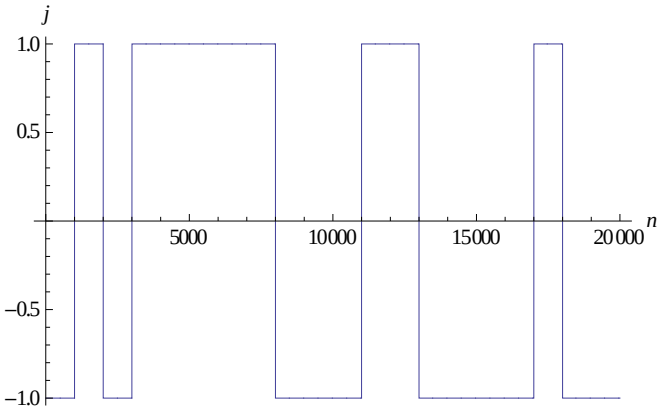
Úkolem této diplomové práce bylo vytvořit program v prostředí Mathematica, který bude šifrovat užitečný signál pomocí CML systémů. Program byl vytvořen pomocí teorie popsané výše (kapitola 7.1). Na obrázku (Obr. 26) můžete vidět napsanou logickou rovnici a OCOML rovnici v jazyce Mathematica.

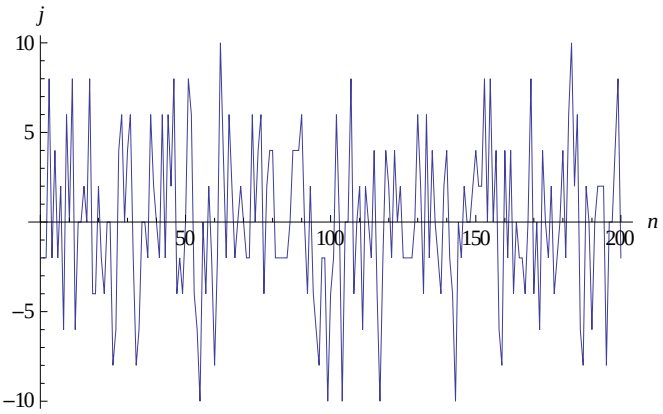
```
Logistic = Compile[{{x, _Real}, {r, _Real}}, r x (1 - x)];


OCOML = Compile[{{x, _Real, 1}, {ϵ, _Real}, {r, _Real}, {L, _Integer},
    {a, _Real, 1}, {idx, _Integer}},
   Flatten[{MapIndexed[
      If[#2[[1]] == 1, a[[idx]],
        (1 - ϵ) Logistic[x[[#2[[1]]]], r] + ϵ Logistic[x[[#2[[1]] - 1]], r]
       ] &, x], idx + 1}]
  ];
```

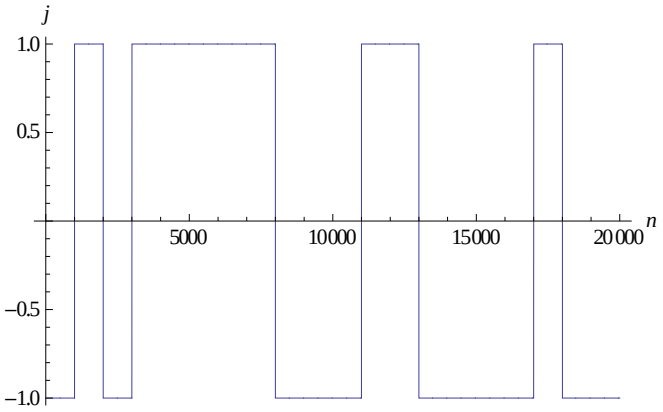
Obrázek 26 Nadefinovaná Logická rovnice a OCOML v jazyce Matematika

Vytvořený program dokáže zašifrovat užitečný signál do nesrozumitelného šumu pomocí CML systémů. V dalším kroku se chaotický šum opět dešifruje na užitečný signál. Výstupem programu je graf OCOML struktury, dále graf originálního signálu, šifrovaného signálu a dešifrovaného signálu.

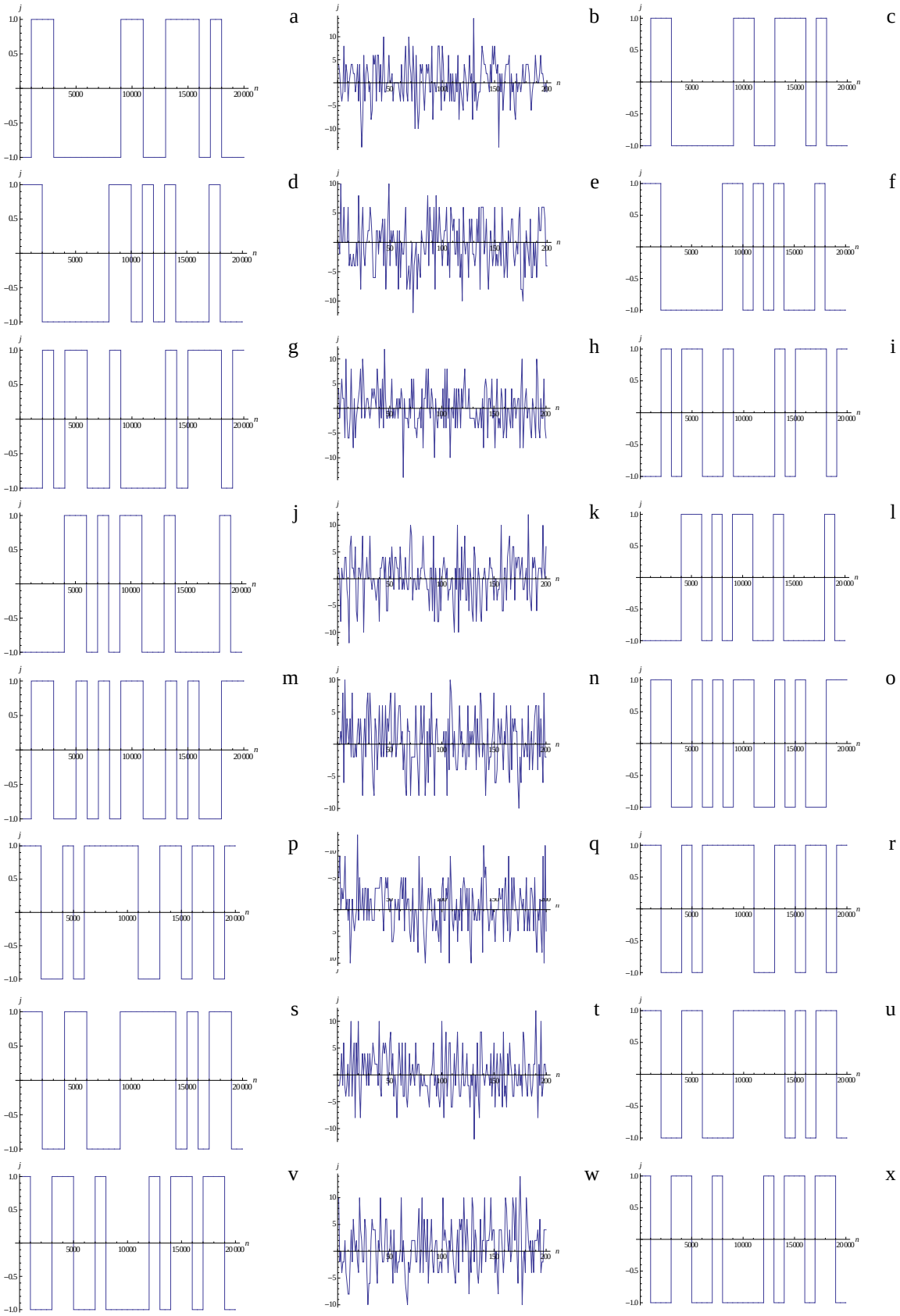








Obrázek 27 Výstup programu šifrování signálu, a) OCOML struktura, b) užitečný signál, c) šifrovaný signál, d) dešifrovaný signál



Obrázek 28 Ukázka šifrování pro 8 různých signálů. Levý sloupec – otevřený text, prostřední – šifrovaný text, pravý – dešifrovaný text.

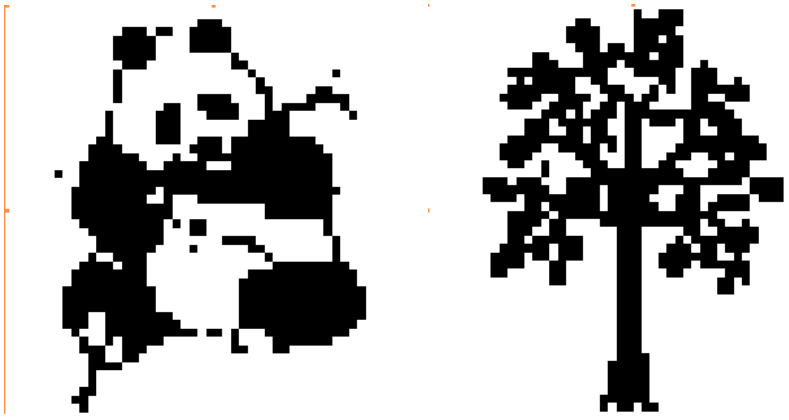
Úspěšnost dešifrování je popsaná v následující tabulce (Tab. 1). Bylo provedeno 10 pokusů šifrování a dešifrování. Z uvedených dat vyplívá, že v průměru dešifrujeme správně 99,9998 % dat, což můžeme považovat za dobrý výsledek.

Tabulka 1 Úspěšnost dešifrování

Číslo pokusu	Úspěšnost v %
1	100
2	100
3	99,9975
4	100
5	100
6	100
7	100
8	100
9	100
10	100
Průměr	99,9998

7.2 Šifrování rastrových obrázků

V této části bude podrobněji rozebráno šifrování jednoduchých obrázků, vyjádřených pouze maticí. Každá matice má velikost 50×50 prvků, kde jeden prvek matice reprezentuje jeden pixel obrázku. Program bude testován s obrázky pandy a stromu (Obr. 29).



Obrázek 29 Obrázek pandy a stromu reprezentované maticí 50×50 bodů

Rastrové obrázky byly vytvořeny následujícím způsobem. Do notebooku v programu Mathematica byl importován obrázek (například panda.gif). Obrázky byl rasterizován a posléze převeden do binární podoby. Výsledkem jsou matice 50×50 prvků, kde hodnota 0 reprezentuje černou barvu a 1 bílou barvu. Ukázka kódu v matematice je na následujícím obrázku (Obr. 30).

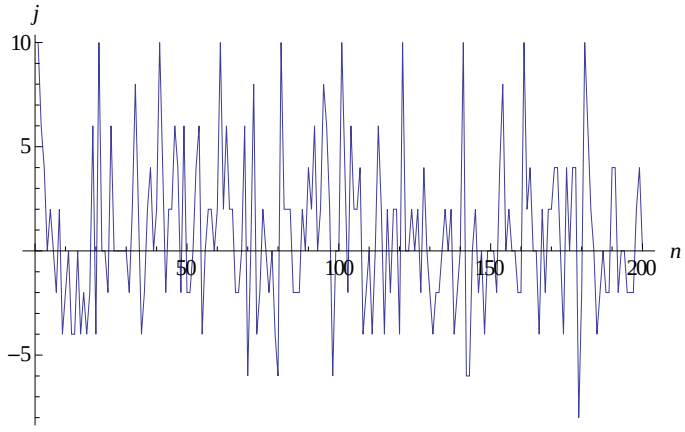
```
obr = Import["ExampleData/panda.gif"];

obr2 = Rasterize[obr, RasterSize → 50, ImageSize → 50];

obr3 = Binarize[obr2];

obr4 = ImageData[obr3];
```

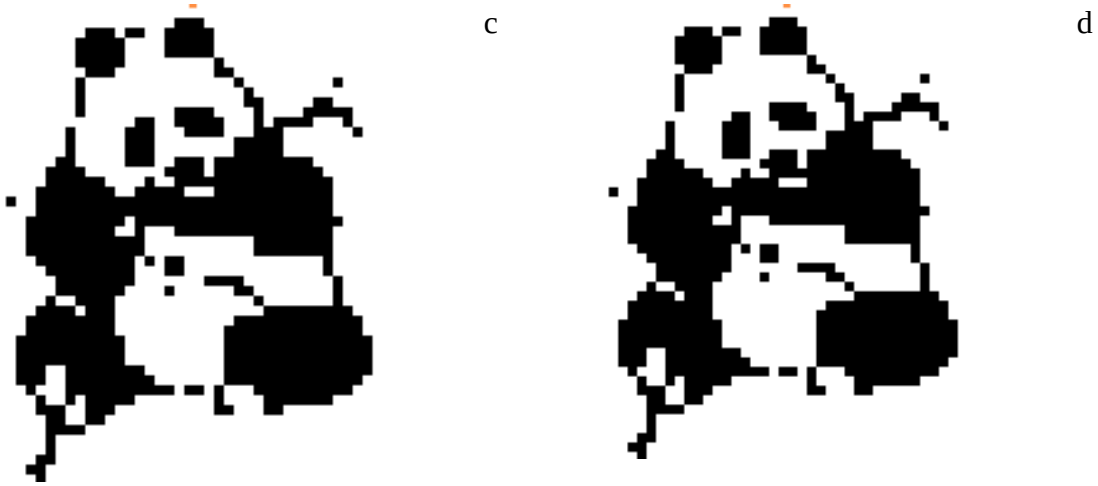
Obrázek 30 Import obrázku panda

Obrázek 31 Originály obrázků před pasterizací

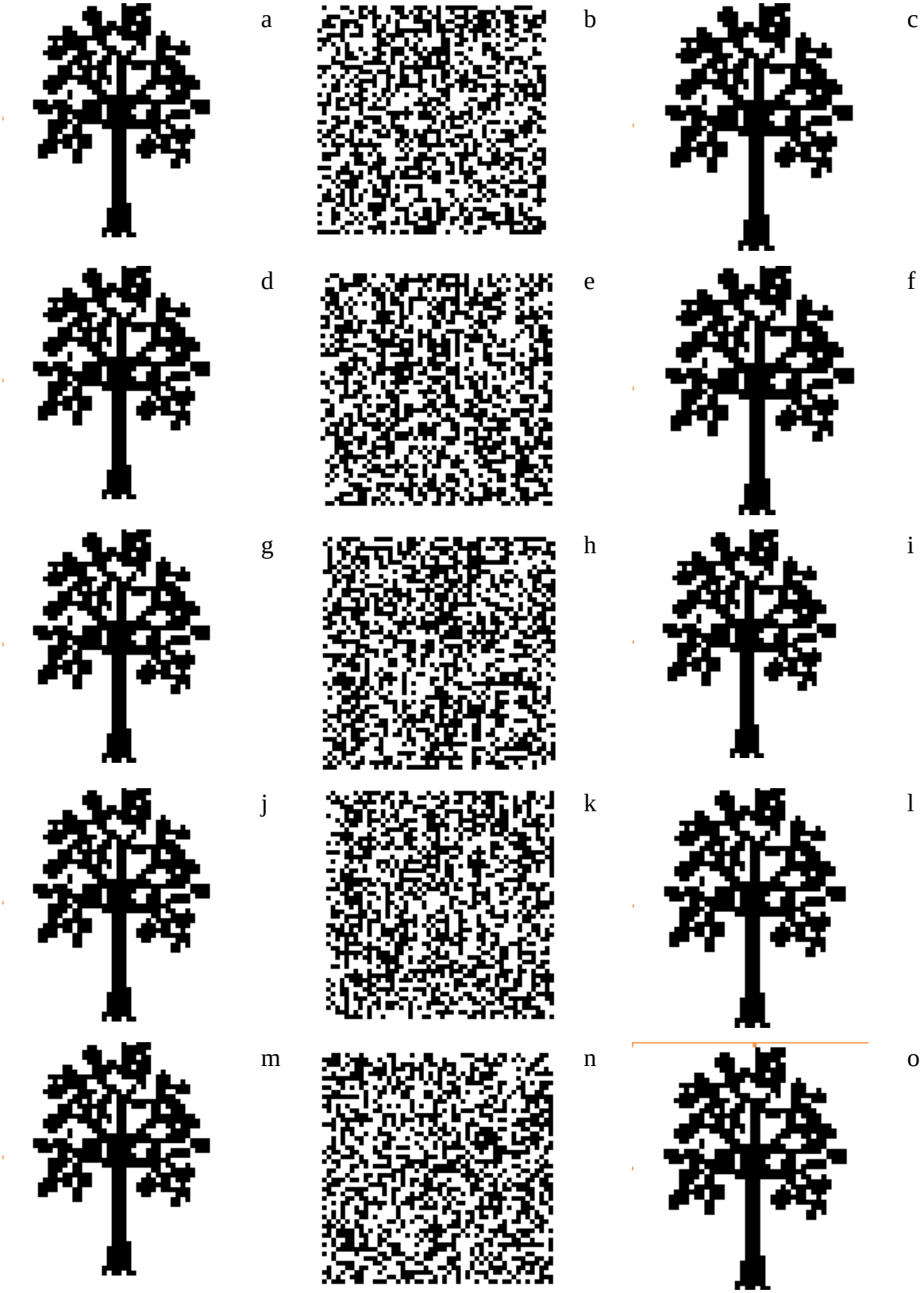
Šifrování matic bude probíhat podobně jako u užitečného signálu. Na začátku je třeba provést modulaci $0 \rightarrow -1$. Dále použijeme rovnici OCRML (9), kde f je logická rovnice s parametrem $r = 4$, $\varepsilon = 0.8$, $L = 60$. Z této rovnice bude získán vektor a_j. Poté bude použita rovnice OCOML (8), kde f představuje logickou rovnici. Proměnná $r = 4$, $\varepsilon = 0{,}95$, $L = 60$.

Dešifrování bude probíhat obdobným způsobem jako v kapitole 7.1. Použije se rovnice (12) a totožné parametry jako při šifrování.

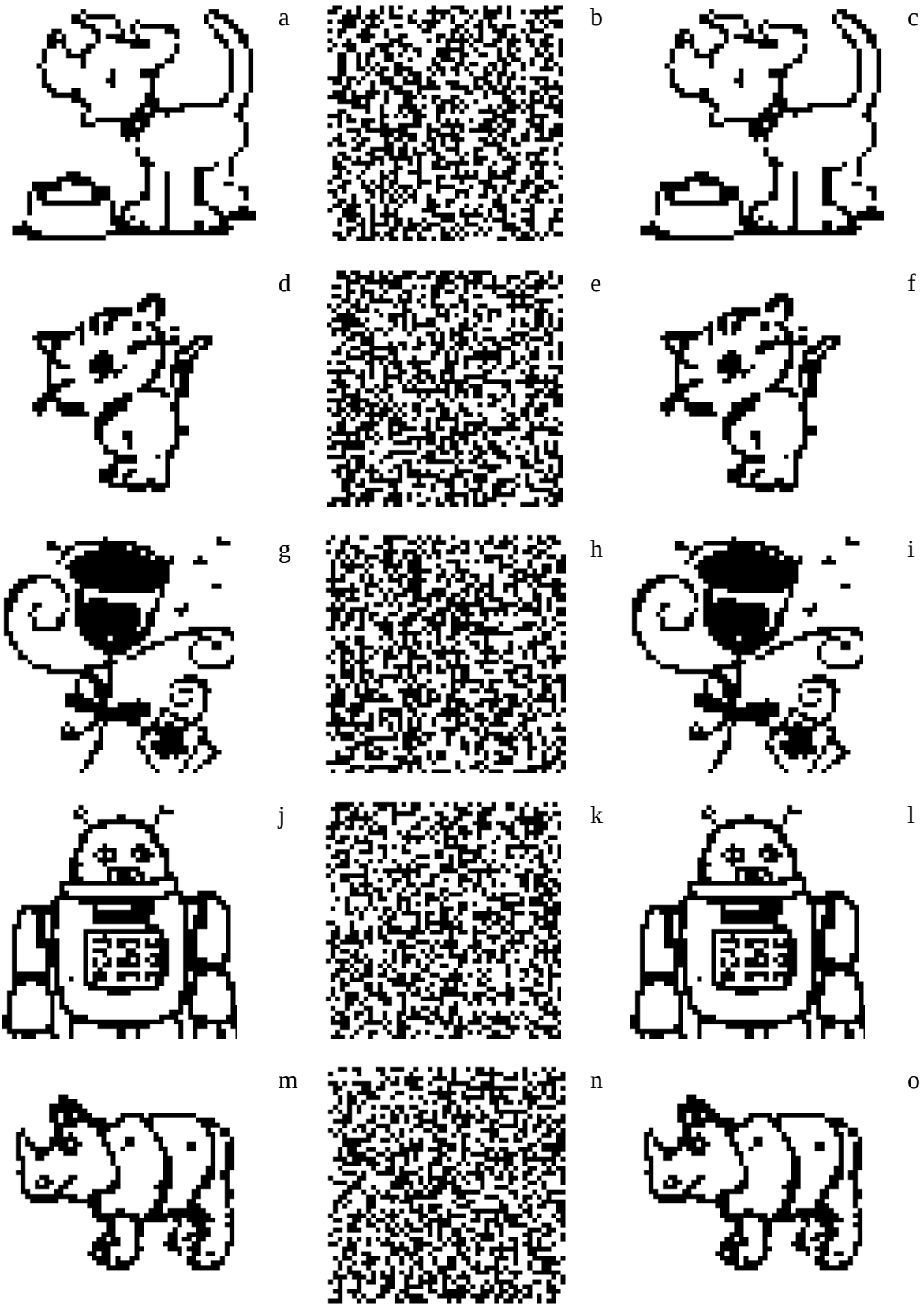




Obrázek 32 Výstup programu šifrování rastrových obrázků, a) šifrovaná data, b) šifrovaná data zobrazená jako obrázek, c) originál obrázku, d) dešifrovaná obrázek



Obrázek 33 Ukázka 5 pokusů šifrování rastrových obrázků. Levý sloupec – originální obrázky, prostřední sloupec – šifrované obrázek, pravý sloupec – dešifrované obrázky.



Obrázek 34 Ukázka šifrování 5 různých rastrových obrázků. Levý sloupec - originální obrázky, prostřední sloupec – šifrovaná data, pravý sloupec- dešifrované obrázky.

Při testování úspěšnosti dešifrování byla získána data uvedená v tabulce (Tab. 2). Šifrování a dešifrování bylo provedeno desetkrát na maticích obrázku panda a strom. Průměrná úspěšnost byla 99,9998%. Z těchto dat je patrné, že při šifrování větších množství dat, než je užitečný signál, je těžší dosáhnout 100% úspěšnosti. Při potřebě zvýšit úspěšnost dešifrování, bude nutno upravit parametry CML systému. Zvýšení účinnosti šifry ovšem přináší i zvýšení časové náročnosti.

Šifrování bylo také vyzkoušeno na 5 dalších rastrových obrázcích (Obr 34), úspěšnost dešifrování byla obdobná jako u předchozích obrázků.

Tabulka 2 Úspěšnost dešifrování

Číslo pokusu	Úspěšnost v %	
	Strom	Panda
1	100	100
2	100	100
3	100	99,9996
4	100	100
5	99,9996	100
6	100	99,9992
7	99,9996	100
8	99,9996	99,9996
9	100	100
10	100	99,9996
Průměr	99,9999	99,9998

7.3 Využití CML pro šifrování obrázků

V této kapitole bude probráno šifrování obrázků pomocí CML systémů. Vytvořená aplikace umí pracovat s obrázky typu JPG, BMP, PNG a PCX.

Před samotným šifrováním musí být provedena modulace dat, tak aby byla vhodná pro šifrování. Nejprve je nutné importovat data o barevných složkách a následně tato data převést do digitální podoby. Ukázka v jazyce mathematica je vidět na (Obr. 35).

```
obrData = Import["ExampleData/obr.jpg", "Data"];
digit = IntegerDigits[obrData, 2, 8];
digit2 = Flatten[digit];
MM = Partition[digit2, Dimensions[digit2][[1]] / 8];
```

Obrázek 35 Import a digitalizace dat obrázků

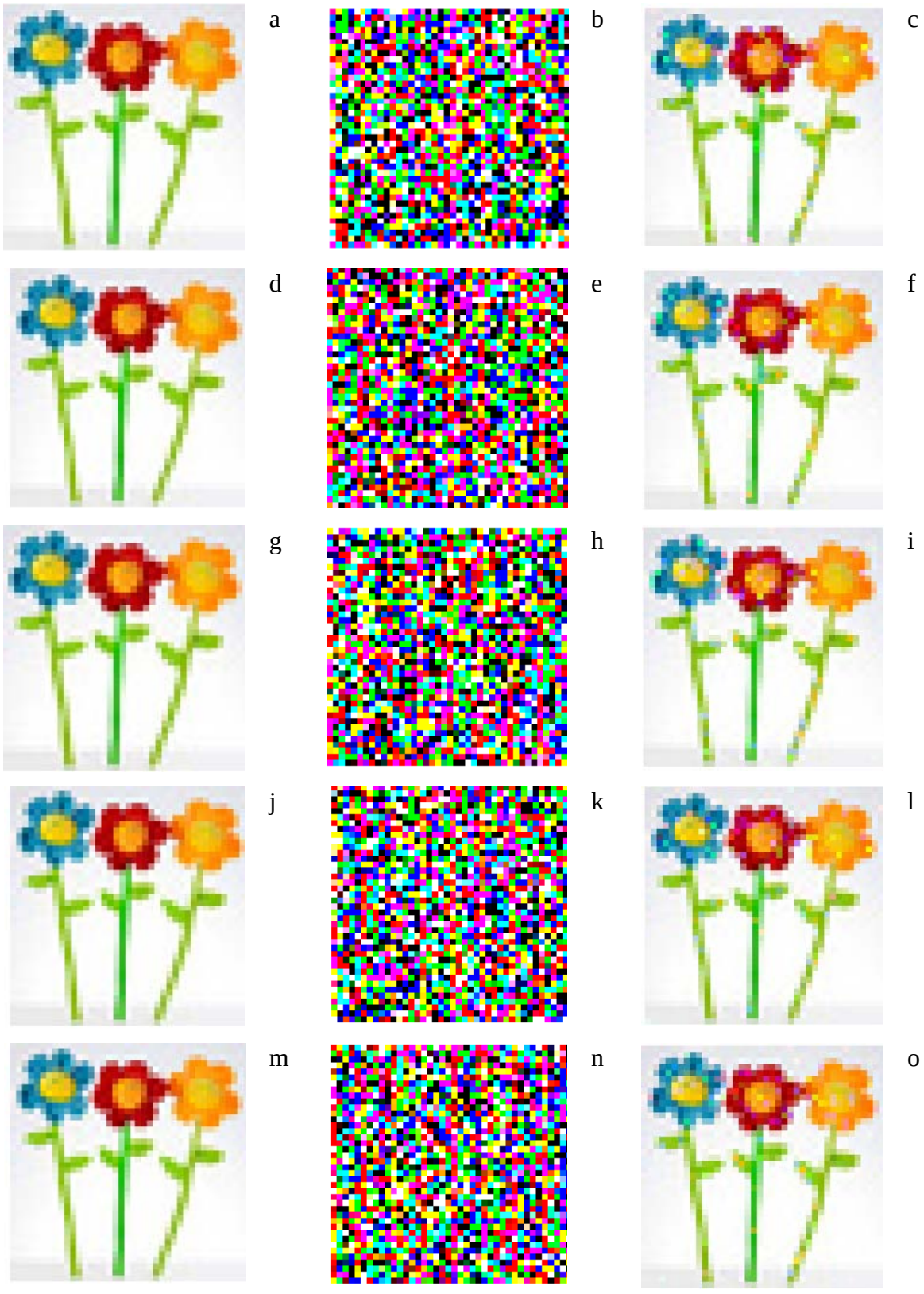
Pro šifrování obrázků bude opět použita rovnice OCOML (8), kde *f* představuje logickou rovnici. Proměnná $r = 4,\ \varepsilon = 0{,}95,\ L = 48$. Vektor a_j bude získán z rovnice (9), kde *f* je logická rovnice s parametrem $r = 4,\ \varepsilon = 0.8,\ L = 60$ Dále bude použita modulace chaotického signálu (10). Šifrovaný text bude získán vynásobením chaotického signálu a užitečného signálu podle rovnice (11).

Dešifrování se provádí podle rovnice (12), kde nejprve bude získán totožný chaotický signál, jako pro šifrování. Dešifrovaný signál se získá po vynásobení šifrovaného signálu a chaotického signálu.

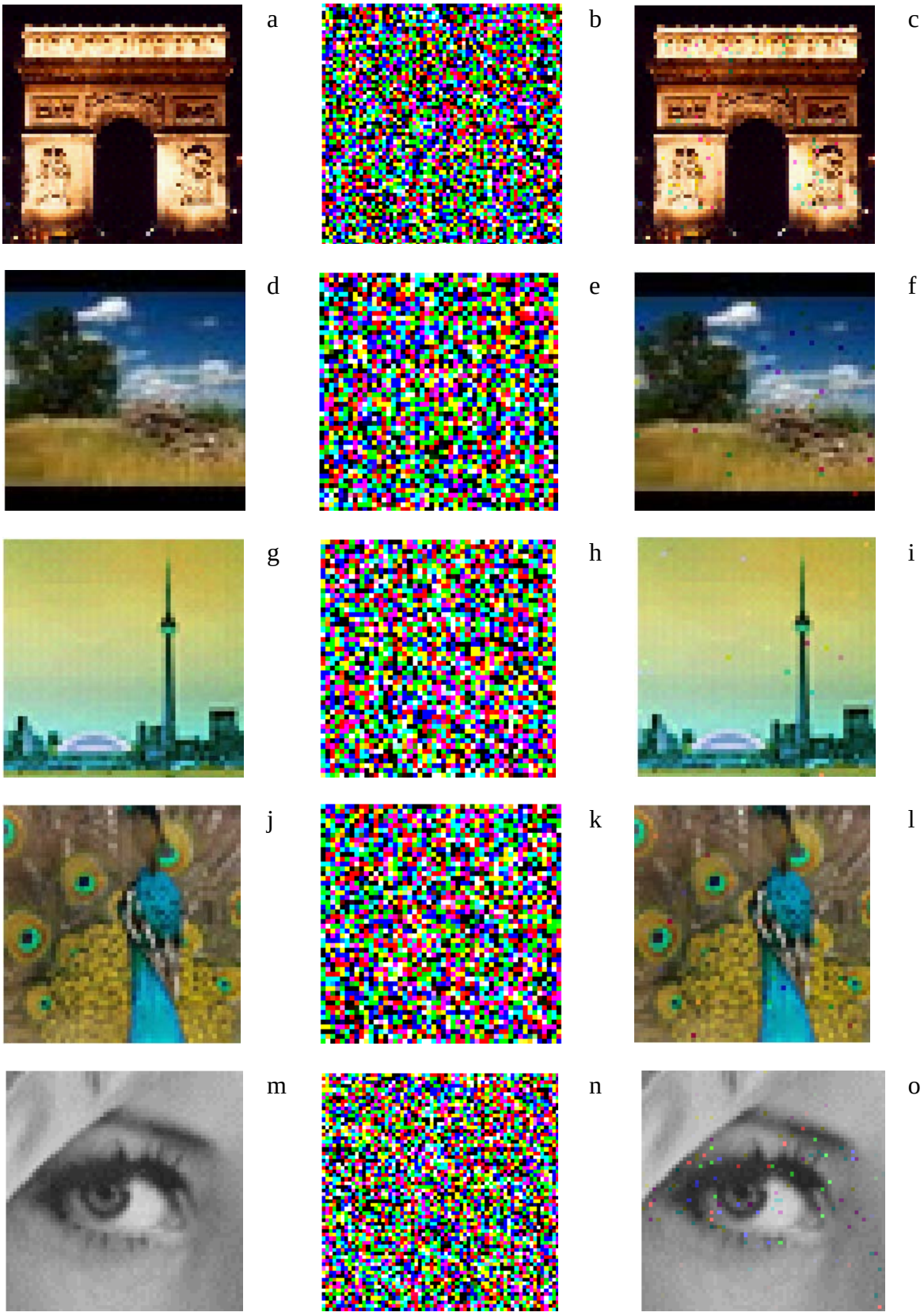
Tabulka 3 Úspěšnost dešifrování

Číslo pokusu	Úspěšnost v %
1	99,9955
2	99,9960
3	99,9950
4	99,9963
5	99,9959
6	99,9958
7	99,9955
8	99,9951
9	99,9957
10	99,9957
Průměr	99,9957

Bylo provedeno 10 pokusů šifrování a dešifrování obrázků. Pro pokus byl použit obrázek kytka.jpg, s rozměry 40x40 pixel. Takto malý obrázek byl použit, protože šifrování obrázků je časově náročnější. Na obrázku (Obr. 36) můžeme vidět 5 pokusů.



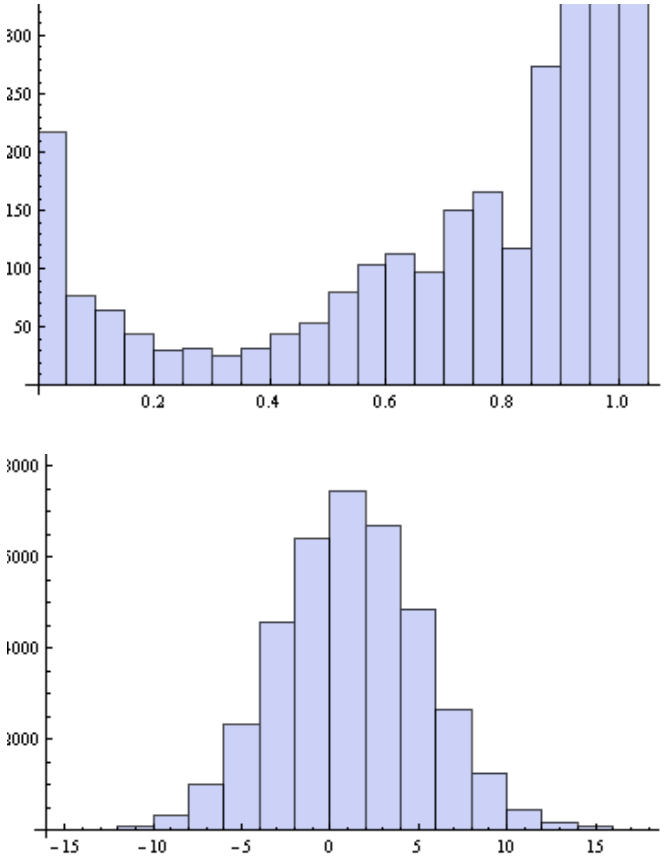
Obrázek 36 Ukázka 5 pokusů šifrování obrázku kytka.jpg. Pravý sloupec - originál obrázku, prostřední sloupec - šifrovaný obrázek, levý sloupec – dešifrovaný obrázek.



Obrázek 37 Ukázka šifrování 5 různých obrázků. Levý sloupec – originální obrázky, prostřední sloupec – šifrované obrázky, pravý sloupec dešifrované obrázky.

Úspěšnost dešifrování je zobrazena v tabulce (Tab. 3). Z naměřených dat vyplívá, že průměrná úspěšnost dešifrování je 99,9957 %. Pokud bychom za každou ceny chtěli dosáhnout 100% úspěšnosti, mohli bychom lépe nastavit parametry pro šifrování, ale časová náročnost programu by byla příliš velká. Poté bylo provedeno 5 pokusů šifrování 5 různých obrázků obdobné velikosti. Úspěšnost dešifrování byla srovnatelná s daty uvedenými v tabulce (Tab. 3).

Na obrázku (Obr. 38) můžeme vidět histogramy obrázku kytka.jpg a šifrovaných dat. Zatím co histogram obrázku má rozdělení odpovídající použitým barvám, histogram šifrovaných dat má Gaussovo rozdělení. Z čehož vyplívá, že touto cestou ze šifrovaných dat nezískáme žádné informace o utajených datech.



Obrázek 38 Histogram a) obrázku kytka.jpg, b) šifrovaných dat

7.4 Vizualizace ve WebMathematice

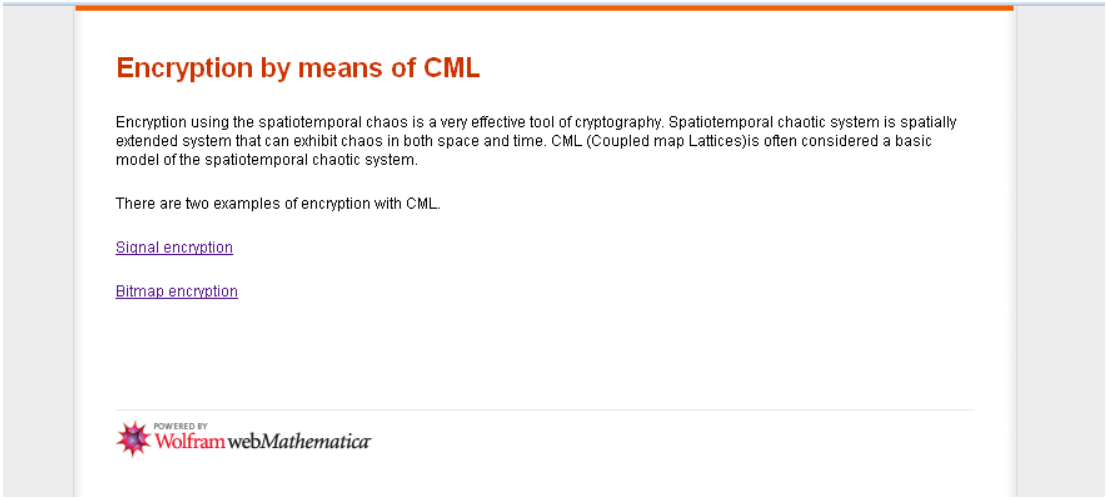
WebMathematica je prostředí které spojuje Mathematicu a technologie webových serverů. Díky tomuto produktu můžeme internetové stránky obohatit o interaktivní výpočty a vizualizace. Zatím co Mathematice je spíše vývojové prostředí, webMathematica je prostředí vhodné pro vizualizaci.

Vytvořené aplikace pro šifrování dat pomocí CML systémů byly převedeny do prostředí webMathematica. Zde jdou dostupné dvě vizualizace vytvořených kryptosystémů:

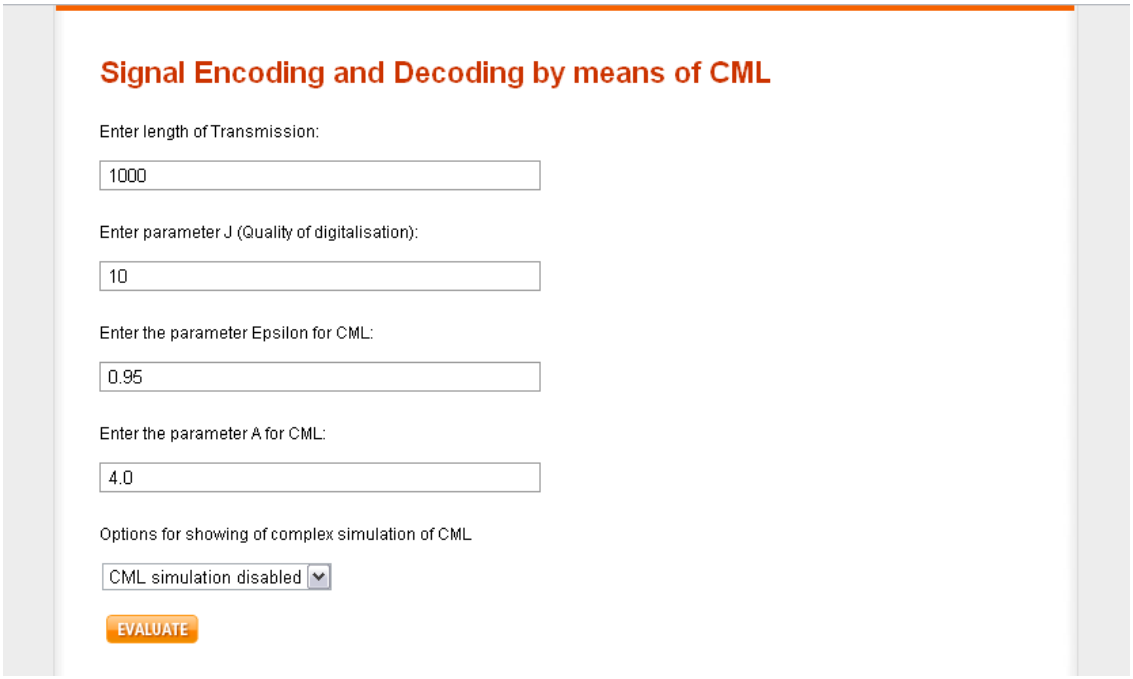
http://mathematica.fai.utb.cz:8080/webMathematica/Eva/cml.jsp

Zde je možné si ověřit funkčnost aplikace vytvořené v Mathematice pro šifrování užitečného signálu a rastrových obrázků. Do programu je možné zadávat proměnné *a, ε, J* a délku signálu. Každý si tak může vyzkoušet vliv těchto proměnných na vznik chaosu a kvalitu šifrování.

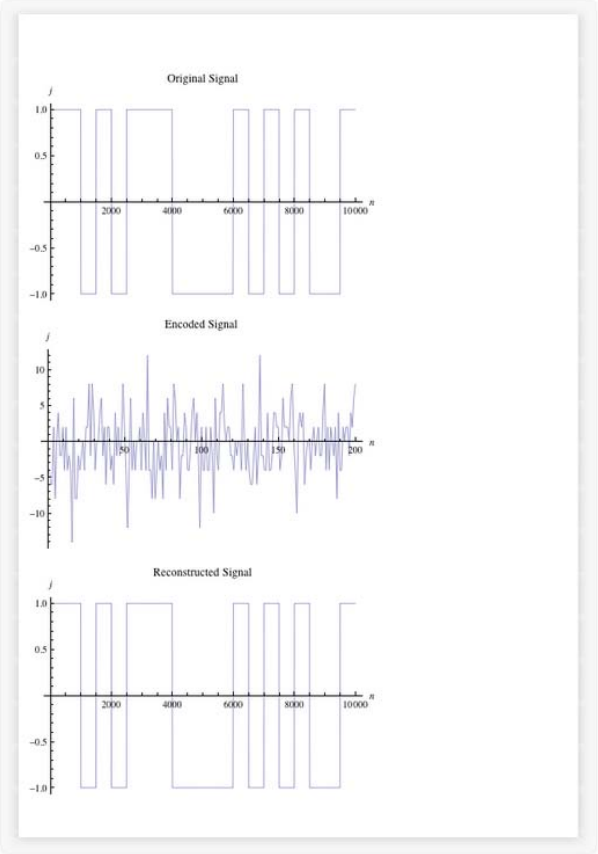
Aplikace pro šifrování obrázků nebyla umístěna na web, jelikož je časově příliš náročná. WebMathematica pro každý dotaz na server vymezí jen krátký čas a proto by tato aplikace nemohla být zprovozněna.



Obrázek 39 Úvodní stránka webových stránek



Obrázek 40 Ukázka webových stránek. Formulář pro zadání parametrů kryptosystému



Obrázek 41 Zobrazení výsledků šifrování užitečného signálu

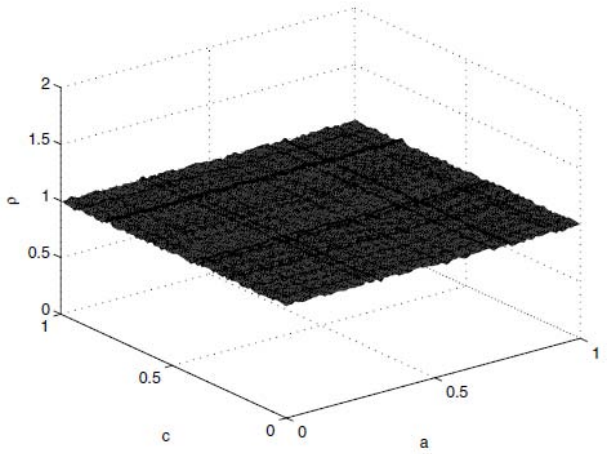
7.5 Možnosti útoků na CML

V této sekci budou popsány některé možné útoky na CML šifry. Bude zde rozebrán útok pomocí chybové funkce, diferenciální útok, útok se znalostí holého textu, útok hrubou silou, a útok s možností volby otevřeného a šifrovaného textu.

7.5.1 Konfuze a difuze

Chcete-li obecně odolat útokům, měl by kód mít dvě základní crypto-grafické vlastnosti, konfuzi a difuzi. Konfuze potlačuje jednotnost všech klíčů. Konfuze je vyhodnocována na základě nezávislosti na pravděpodobnosti rozdělení šifrovaného text pro daný klíč. Matematicky je vyjádřena jako $\rho\ (c \mid a)\ (c = G/2^{32},\ G = G_n\ (j;a),\ a = aI)$, která je zobrazena na obrázku (Obr. 42). Podmíněné rozdělení pravděpodobností šifrovaného textu je shodné pro všechny klíče. Tudíž je zaručena konfuze šifry.

Difuze odráží silnou citlivost klíče na drobné změny. V našem případě je klíč dokonce citlivý na velmi malé změny 2^{-47}, což potvrzuje difuzi šifry [8].

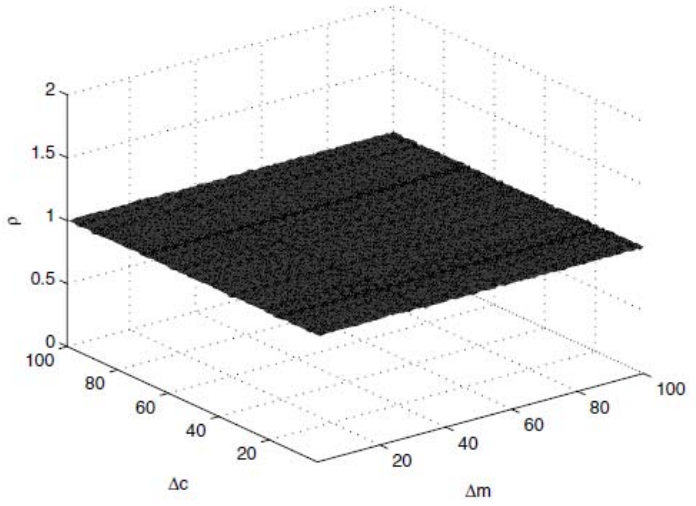


Obrázek 42 Podmíněná pravděpodobnost rozdělení p (c|a)

7.5.2 Diferenciální útok

K útoku na kryptosystém je možné využít některých diferenciál vztahů mezi šifrovaným textem a dešifrovaným. Například některé nedokonalé statistické vlastnostmi výstupního datového proudu mohou být použity k prolomení šifry. Chceme-li zjistit, zda v kódu existují takovéto diferenciální vztahy, musíme prověřit podmíněné pravděpodobnosti

šifrovaného textu $\rho\ (\Delta c \mid \Delta m)\ (\Delta c = \Delta G\ /2^{32},\ \Delta G = Gn\ (j;\ M'^{j}_{n})\ -\ Gn\ (j;\ M^{j}_{n}))$ za podmínky $\Delta m = \Delta\ M^{j}_{n}/2^{32}\ (\Delta M^{j}_{n} = M'^{j}_{n} - M^{j}_{n})$. Na obrázku (Obr. 43) můžeme pozorovat, že podmíněná pravděpodobnost je stejná pro všechny hodnoty, tudíž kryptosystém je imunní vůči diferenciálnímu útoku [8].



Obrázek 43 Podmíněná diferenciální pravděpodobnost ρ (Δc|Δm)

7.5.3 Útok se znalostí otevřeného textu

Pokud útočník zná kompletní informace o šifře a její realizaci, pak může odhalit klíč díky zveřejnění šifrovaného textu a známého otevřeného textu. Útok je veden pomocí inverzních analytických výpočtů otevřeného textu a šifrovaného textu, tedy $S_n j$. Obtížnost takovéhoto útoku může být vyjádřena následovně: $x_{n+1}^{j} = (1 - \varepsilon)\ f\ (x_n^{j}, a_n^{j}) + \varepsilon\ f\ (x_n^{j}\text{-}1, a_n^{j-1})$ a $f\ (x_n^{j},\ a_j) = (3.9 + 0.1 a_j)\ x_n^{j}\ (1 - x_n^{j})$.

Typický útok se znalostí otevřeného textu je útok hrubou silou, kde kód je napaden pomocí zkoušení všech možných klíčů. V našem případě je množina všech klíčů rovna 2^{47L} [8].

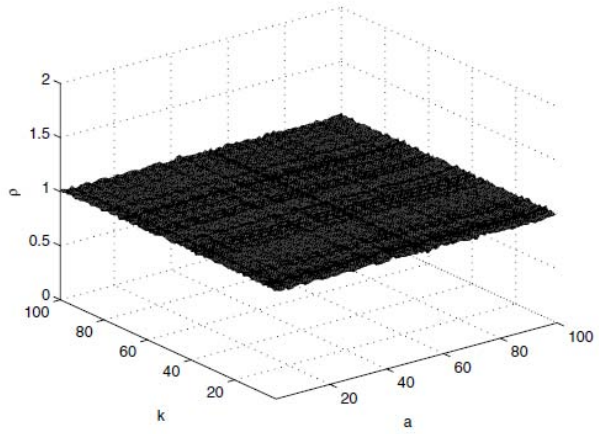
7.5.4 Útok s možností volby otevřeného textu a šifrovaného textu

Při tomto útoku si útočník vybere některé speciální otevřené a šifrované texty, aby získal specifické klíčové vektory. V případě, že některé klíčové vektory odpovídají určitým klíčům, tj. existují některé charakteristické vztahy mezi klíčovými vektory a klíči, pak tyto útoky mohou být účinné.

Vztah mezi klíčovými vektory s $(=S_n^{j}\ /2^{32})$ a klíči $a(a = aI)$ je zjišťuje pomocí rozdělení pravděpodobností $\rho\ (s|a)$ (Obr. 44). Uvádí se, že vlastnosti klíče nelze získat z žádného

klíčového vektoru. Tedy žádný zvláštní otevřený nebo šifrovaný text nelze vybrat pro prolomení klíče. Jinými slovy, útok s možností volby otevřeného / šifrovaného textu má stejnou účinnost jako útok se znalostí holého textu této šifry.

Souhrnně lze říci, útok se znalostí holého textu je nejefektivnější útok a jeho nároky pro prolomení kódu jsou 2^{40L}. Mimo to lze bezpečnost šifry pohodlně zvýšit pomocí přidání mřížek v CML systému [8].



Obrázek 44 Podmíněná pravděpodobnost ρ (s|a)

ZÁVĚR

Hlavním cílem této práce bylo přiblížit problematiku deterministického chaosu a popsat jeho využití pro utajenou komunikaci. Kryptografie je dnes velice žádaným a potřebným oborem. Můžeme se s ní setkávat takřka denně. Pro všechny, kteří komunikují přes internet, využívají internetové bankovnictví, nebo si chtějí bezpečně zálohovat data, je tento obor velice důležitý. V první části této práce byly stručně popsány základní vlastnosti kryptografie.

V další části byla vylíčena historie deterministického chaosu a její vlastnosti. Chaos je poměrně nový vědní obor, který se začal prosazovat až v polovině 20. století. Dnes už má své důležité uplatnění v mnoha odvětvích, nicméně v některých odvětvích, jako je třeba kryptografie, se stále neprosadil.

Časoprostorový chaos je odvětvím deterministického chaosu. Tento systém dokáže šířit chaotické chování jak v prostoru, tak v čase. Tato práce se snažila využít vlastnosti časoprostorového chaosu pro utajenou komunikaci. Byl navržen nástroj pro šifrování užitečného signálu, rastrových obrázků a obrázků typu jpg. V prostředí webMathematica byla vytvořena vizualizace šifrování užitečného signálu a obrázků.

Součástí práce je i prezentace o časoprostorovém chaosu a jeho využití pro utajenou komunikaci. Ta může být použita pro výukové potřeby na Fakultě aplikované informatiky UTB ve Zlíně.

V poslední části práce byla zhodnocena bezpečnost šifrování pomocí časoprostorového chaosu. Tyto kryptosystémy jsou velmi odolné vůči mnoha známým útokům a rozhodně by si zasloužily větší uplatnění na poli kryptografie. Jedna z významných předností chaosu však může být i jeho nevýhodou. Velká citlivost na počáteční podmínky může činit problémy, například pokud se bude aplikace používat na jiném hardwaru. Proto se na tomto oboru bude muset odvést ještě kus práce, než bude opravdu masově použitelný.

ZÁVĚR V ANGLIČTINĚ

The main objectives of this work were approximate deterministic chaos and describe its use for secret communications. Cryptography is highly sought and needed discipline now. We can meet with her almost daily. For all people who communicate over the internet, using internet banking, or want to securely back up data, this field is very important. In the first part of this work were briefly describes the basic of cryptography.

In another part in this work was portrayed history of deterministic chaos and its properties. Chaos is a relatively new science, which he started making up in the mid-20th century. Today it has important applications in many sectors, but in some sectors, such as the cryptography is still not taken up.

Spatiotemporal chaos is a branch of deterministic chaos. This system can distribute the chaotic behavior in both space and time. This work has sought to exploit properties of space-time chaos for secret communication. Instrument was designed to encrypt of signal, bitmaps and JPG files. In the environment webMathematica was created visualization encryption useful signal and images.

The work also includes the presentation of spatiotemporal chaos and its applications for secret communication. It can be used for instructional in the Faculty of Applied Informatics, Tomas Bata University in Zlín.

In the last part of this work was evaluated the safety of the spatiotemporal chaos encryption. These cryptosystems are very resistant to many known attacks, and certainly deserve more application in the field of cryptography. One major advantage of the chaos, however, may be also its disadvantage. Large sensitivity to initial conditions may be problems, for example, if an application will use on different hardware. Therefore, in this field will have to make a piece of work than is really mass-usable.

SEZNAM POUŽITÉ LITERATURY

[1] GONZALES-MIRANDA, J. M., Synchronization *And Control Of Chaos: An Introduction For Scientists And Engineers*. World Scientific Publishing Company, 2004. 224 s. ISBN 978-1860944888.

[2] HOSTE, Jim. *Mathematica DeMYSTiFied*. McGraw-Hill Professional, 2008. 408 s. ISBN 978-0071591447.

[3] RUSKEEPAA, Heikki. *Mathematica Navigator: Mathematics, Statistics and Graphics, Third Edition*. Academic Press, 2009. 1136 s. ISBN 978-0123741646.

[4] BENERJEE, Santo. *Chaos Synchronization and Cryptography for Secure Communications: Applications for Encryption*. Information Science Publishing, 2010. 350 s. ISBN 978-1615207374.

[5] GLEICK, James. *Chaos: vznik nové vědy*. Ando Publishing, 1996. 350s. ISBN 80-86047-04-0.

[6] PRIGOGINE, Ilya. *Řád z chaosu: Nový dialog člověka s přírodou*. Mladá fronta, 2001. 320 s. ISBN 80-204-0910-6.

[7] HORÁK, Jiří. *Deterministický chaos a jeho fyzikální aplikace*. Academia, 2003. 437 s. ISBN 8020009108.

[8] ZELINKA, Ivan. *Evolutionary Algorithms and Chaotic Systems*. Springer, 2010. 560 s. ISBN 9783642107061

[9] H.G. Schuster (eds), *Handbook of Chaos Control*, Wiley-Vch, 1999, ISBN 3-527-29436-8

[10] JAŠEK, Roman. *Informační a datová bezpečnost*. UTB, 2006. 140 s. ISBN 80-7318-456-7

[11] ZELINKA, Ivan. *Řízení deterministického chaosu.* [online]. [cit. 2010-06-03]. Dostupné z WWW: <://www.fai.utb.cz/people/zelinka/chaos/Chaos_control.html>.

[12] URUBA, Václav. *Náhoda v exaktní vědě*. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://www.it.cas.cz/~uruba/docs/lit/Nahoda.pdf>

[13] HECZKO, Stanislav. Teorie chaosu a chování otevřených systémů. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://www.sds.cz/docs/prectete/epubl/she_tch.htm>

[14] *Teorie chaosu*. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Teorie_chaosu>

[15] *Kvantová kryptografie*. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Kvantová_kryptografie>

[16] *Henri Poincaré*. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Henri_Poincaré>

[17] *Edward Lorenz*. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Edward_Lorenz>

[18] Mitchell Feigenbaum. [online]. [cit. 2010-06-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Mitchell_Feigenbaum>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CML	Coupled map lattices
CS	Complete Synchronization - úplná synchronizace
CSK	Chaotic shift keying -chaotické klíčování
DCSK	Differential chaos shift keying - diferenční chaotické klíčování
LGS	Linear Generalized Synchronization - lineární synchronizace
LS	Lag Synchronization, zpětnovazební synchronizace
OCOML	One-way coupled open map lattice
OCRML	One-way coupled ring map lattice
PS	Phase Synchronization - fázová synchronizace

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: ZDROJOVÝ KÓD ŠIFROVÁNÍ UŽITEČNÉHO SIGNÁLU

```
(*parametry*)
T = 50;
delkatransmise = 1000;
T = 50;
J = 20;
Signalu = 20;

(* ------ OCRML sn sekvence -------*)

Logistic = Compile[{{x, _Real}, {A, _Real}}, A x (1 - x)];
SPL = Compile[{{x, _Real,
     1}, {ε, _Real}, {A, _Real}, {L, _Integer}},
   MapIndexed[
    If[#2[[1]] ==
       1, (1 - ε) Logistic[x[[#2[[1]]]],
         A] + ε Logistic[x[[L]], A],
      (1 - ε) Logistic[x[[#2[[1]]]],
         A] + ε Logistic[x[[#2[[1]] - 1]], A]
      ] &, x]
   ];
Subscript[x, Start] = Table[Random[], {i, 60}];
ocoml = NestList[
   SPL[#1, .8, 4, Dimensions[Subscript[x, Start]][[1]]] &,
Subscript[
   x, Start], delkatransmise];
(*Print[ListDensityPlot[Transpose[ocoml],Mesh->False,AspectRatio-
>.3]]\;*)

Subscript[s, n] = Take[Transpose[ocoml][[1]], delkatransmise];
(*Print[ListPlot[Subscript[s, n],PlotJoined->True]];*)

(*------ OCOML kanaly ----*)

SPL = Compile[{{x, _Real,
     1}, {ε, _Real}, {A, _Real}, {L, _Integer}, {sn, _Real,
     1}, {idx, _Integer}},
   Flatten[{MapIndexed[
      If[#2[[1]] == 1, sn[[idx]],
```

```
        (1 - ε) Logistic[x[[#2[[1]]]],
           A] + ε Logistic[x[[#2[[1]] - 1]], A]
        ] &, x], idx + 1}]
   ];

εε = .95;
AA = 4;
sites = 60;

spch = NestList[
   SPL[Take[#1, 60], εε, AA,
     Dimensions[Subscript[x, Start]][[1]], Subscript[s, n],
     Take[#1, -1]] &, Flatten[{Subscript[x, Start], 1}],
   delkatransmise];

Print[ListDensityPlot[Take[Transpose[spch], 60, delkatransmise],
   Mesh -> False, AspectRatio -> .3]];
canal1 = Take[Transpose[spch], 60, delkatransmise];
canal2 = Table[canal1[[i]], {i, 1, 60, 3}];

(*----- Signal -----*)

S = Flatten /@ ((RealDigits[#1, 2, J][[1]] & /@ #1 & /@
       canal2) /. {0 -> -1});
MM = Table[
    Random[Integer, {0, 1}], {Signalu}, {i,
     Dimensions[S][[2]]/T/J}] /. {0 -> -1};
M = Table[
   Flatten[Table[MM[[k, i]], {i, Dimensions[MM][[2]]}, {j, T J}]],
  {k,
     Signalu}];

(*  Transmise 2D  *)

G = M*S;
(*Print[ListPlot[Take[G[[1]],100],PlotJoined->True]];*)
Gtotal = Plus @@ G;
Print[ListPlot[Take[Gtotal, 200], PlotJoined -> True,
   AxesLabel -> {n, j}]];

```

```
(*  Decode  *)
mx1 = Gtotal*#1 & /@ S;
mpx = Partition[#1, J] & /@ mx1;
sumaJ = ((Plus @@ #1 & /@ #1) & /@ mpx) // N;
sumaJ1 = Partition[#1, T] & /@ sumaJ;
sumaN = ((Plus @@ #1 & /@ #1) & /@ sumaJ1)/(T);
Mrek1 = Sign /@ sumaN;

MMM = Table[
   Flatten[Table[
     Mrek1[[k, i]], {i, Dimensions[Mrek1][[2]]}, {j, T J}]], {k,
    Signalu}];

Print[ListPlot[M[[1]], PlotJoined -> True, AxesLabel -> {n, j}]];
Print[ListPlot[MMM[[1]], PlotJoined -> True, AxesLabel -> {n, j}]];

distance =
  Table[HammingDistance[M[[i]], MMM[[i]]], {i, Dimensions[M][[1]]}];
uspesnost =
  100 - (Total[distance]/(Dimensions[M][[1]]*Dimensions[M][[2]]));
Print["uspesnost ", N[uspesnost], " %"];
```