Kvantový šumátor a jeho testování

Hrubý, Jaroslav
2005

Dostupný z http://www.nusl.cz/ntk/nusl-34210

Institute of Computer Science
Academy of Sciences of the Czech Republic

Kvantový šumátor a jeho testování

J. Hrubý and L. Andrej

Technical report No. 948

November 2005

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: (+420)266052085, fax: (+4202) 86 585 789,
e-mail:andre@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Kvantový šumátor a jeho testování[1]

J. Hrubý[2] and L. Andrej[3]

Technical report No. 948

November 2005

Abstrakt:

Je popsán kvantový šumátor jako kvantově-mechanický generátor náhodných čísel. Detailně je popsáno fyzikálně-teoretické řešení takového generátoru, jakož i jeho experimentální realizace. V další části jsou pak testovány jeho vlastnosti ve formě teoretických a empirických testů. Analyzovány jsou také kryptografické testy (normy). Po zhodnocení testů jsou v závěru zmíněny tzv. nelineární metody. V práci jsou také zmíněny možnosti využití kvantového šumátoru v kryptografii.

Keywords:
generátor náhodných čísel, fyzikální realizace, kvantový generátor šumu, generace dat, experimentální výsledky, nehardwarové zdroje náhodnosti, testování vlastnosti generátoru

[1]Projekt registrační číslo 1ET300100403 Aplikace kvantové informatiky na bezpečnost PKI (Infrastruktury s veřejným klíčem)
[2]Fyzikální ústav AV ČR, Na Slovance 2, 182 21 Praha 8, e-mail: hruby.jaroslav@iol.cz
[3]Ústav informatiky AV ČR, Pod Vodárenskou věží 2, 182 07 Praha 8

KVANTOVÝ ŠUMÁTOR A JEHO TESTOVÁNÍ

Projekt registrační číslo **1ET300100403:**

APLIKACE KVANTOVÉ INFORMATIKY
NA BEZPEČNOST PKI
(Infrastruktury s veřejným klíčem)

Řešitel: RNDr. Jaroslav Hrubý, CSc.
Spoluřešitel : RNDr. Ladislav Andrej, CSc.

Fyzikálně-teoretické řešení kvantového šumátoru

1.1 Úvod

Fyzikální generátor náhodných čísel může být založen na nejrůznějších fyzikálních procesech. Jde přitom o to, aby proces samotný byl náhodný ve smyslu nepředpověditelnosti výsledku jeho individuální realizace a vzájemné nekorelovanosti takovýchto individuálních realizací. Tato náhodnost může být:

- praktická, kdy systém je sice po teoretické stránce považován za deterministický, ale je popsán mnoha (často neúplně známými) parametry a obvykle není přesně znám jeho počáteční stav (nebo je technicky obtížné jej připravit opakovaně ve stejném počátečním stavu, příkladem takového generátoru náhodných čísel je třeba ruleta), dalším příkladem může posloužit tzv. deterministický chaos, který je generovaný nelineárním disipativním dynamickým systémem, někdy se hovoří také o kvasináhodném procesu,
- fundamentální, kdy náhodnost je zahrnuta přímo ve fyzikální podstatě jevu a jev je jako náhodný popsán i fyzikálními zákony.

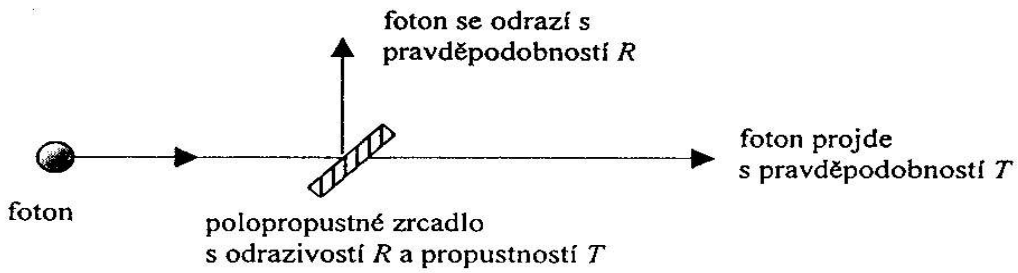
V oblasti kvantové fyziky existuje celá řada jevů, které jsou náhodné ze své samotné podstaty. Zákony kvantové fyziky popisují chování souborů kvantových objektů, nejsou ale, vyjma speciálních případů, schopny předpovědět s určitostí chování individuálního kvantového objektu; předpovídají pouze pravděpodobnosti, s jakými nastane ten či onen konkrétní jev. Důvodem přitom není, jak dnes dotvrzuje velké množství experimentálních dat, momentální neznalost jakýchsi „skrytých“ proměnných – jde o skutečně fundamentální vlastnost mikrosvěta, kterou nelze nijak obejít! To poskytuje velmi dobrý prostor pro konstrukci generátoru náhodných čísel splňujícího nejpřísnější kritéria vyžadovaná právě např. kryptologickými aplikacemi.

Vybereme-li nějaký elementární kvantový proces, který jsme schopni dobře teoreticky analyzovat, tj. určit pravděpodobnosti všech jeho možných výsledků, a jsme-li schopni takový proces opakovaně za dobře definovaných podmínek realizovat, můžeme jej použít jako základ pro generaci náhodných čísel.

1.2 Zvolený fyzikální proces

1.2.1 Dělení světla na děliči svazku

Jedním z elementárních kvantových procesů je dopad světelného kvanta – fotonu na tzv. dělič svazku. Jedná se o zařízení, které se v klasické optice používá k rozdělení jednoho svazku světla na svazky dva. Může jít např. o tzv. polopropustné zrcadlo, existuje však i řada jiných konkrétních realizací tohoto prvku. Zajímavé pro nás je, že snižujeme-li intenzitu světla, začne se projevovat jeho kvantový charakter. Světelná energie se totiž šíří v malých nedělitelných dávkách – fotonech. Dopadne-li jediný foton na dělič svazku, nemůže se rozpůlit; může prostě jen „zvolit“ jednu ze dvou možných cest. Foton ale není kulečníková koule, to, kterou cestou se vydá, je náhodný jev v nejryzejším smyslu.



Takový proces poskytuje několik výhod:

1. jde o jednoduchý, dobře teoreticky popsatelný proces, který poskytuje dvě hodnoty možných výsledků při dopadu jednoho světelného kvanta na dělič svazku,
2. jde o proces dobře experimentálně realizovatelný (dílčí problémy s jeho realizací jsou popsány dále) a kontrolovatelný,
3. výsledek procesu je snadno detekovatelný a dobře převoditelný do elektronické formy k dalšími zpracování,
4. technologie k praktické realizaci je dostupná na komerční úrovni.

Přes jednoduchost zvoleného procesu vyžaduje jeho laboratorní realizace s ohledem na předpokládané aplikace řešení několika dílčích problémů. Ty budou popsány v následujících odstavcích. Do hry v celém experimentálním řetězci vstupuje mnoho dalších náhodných procesů, jejichž vlastnosti nejsou přesně známy. Jejich vliv by tedy měl být eliminován nebo aspoň omezen na minimum, aby výsledky generátoru vycházely z jediného, dobře kontrolovaného a z fyzikálního hlediska fundamentálního náhodného procesu.

1.2.2 Příprava počátečních stavů

Především v současné době není znám způsob, jak opakovaně a kontrolovatelně připravovat jednofotonové stavy světla. Poměrně snadno lze však vytvořit jejich napodobeninu v podobě silně zeslabených laserových pulsů, které vykazují poissonovskou statistiku v počtu fotonů obsažených v pulsu, tj. pravděpodobnost, že puls obsahuje n fotonů lze psát jako:

$$p(n) = \frac{\alpha^n}{n!} e^{-\alpha},$$

kde α je střední počet fotonů v pulsu. Tedy např. pro α = 0,1 dostáváme pravděpodobnosti:

$$p(0) = 90{,}5\ \%,$$
$$p(1) = 9{,}0\ \%,$$
$$p(>2) = 0{,}5\ \%,$$

neboli devět z deseti pulsů neobsahuje žádný foton a přibližně každý dvacátý neprázdný puls obsahuje více než jeden foton. Případy, kdy nedojde k detekci na žádném z výstupů nebo kdy dojde k detekci na obou výstupech, nejsou pro generaci náhodných čísel použitelné. Vzniká tedy otázka, **jaká intenzita (střední počet fotonů) vstupního stavu je optimální.** Pravděpodobnost, že detektor nezaregistruje žádný foton, je $p_0=e^{-\alpha/2}$ (je-li dělič vyvážený

[*R=T],* pak na obou jeho výstupech je střední počet fotonů poloviční než byl na vstupu). Tedy pravděpodobnost, že nebude detekován foton ani na jednom výstupu $p_{00} = p_0^2 = e^{-\alpha}$, pravděpodobnost, že bude detekován foton na obou výstupech $p_{DD} = (1-p_0)^2 e^{-\alpha/2}$.

Pak pravděpodobnost, že puls neposkytne výsledek použitelný pro generaci náhodných čísel, $p_{00} + p_{DD}$, nabývá minimální hodnoty pro střední počet fotonů vstupního pulsu $\alpha = 2 \ln 2 = 1{,}39$ fotonu na puls. V tom případě bude puls použitelný s pravděpodobností $1 - p_{00} - p_{DD} = 0{,}5$.

Dalším důležitým problémem je **otázka vzájemné nekorelovanosti individuálních realizací** tohoto jevu. Z hlediska fyzikálního popisu se předpokládá, že mezi následnými individuálními realizacemi je celý experimentální systém uveden do téhož počátečního stavu, který je zcela nezávislý na předchozích realizacích jevu. To lze velmi dobře předpokládat u samotného děliče svazku, který představuje klasické (makroskopické) zařízení, které vystupuje v celém procesu jen ve funkci parametru, předpokládá se, že interakce s fotonem dělič téměř neovlivňuje a změna jeho stavu je zanedbatelná. Pokud jde o vlastnosti zdroje fotonů, lze pro naši praktickou realizaci rovněž předpokládat, že jeho vlastnosti jsou na časových škálách generace fotonů (10^{-4}-10^{-15} s) konstantní. Máme na mysli zejména střední frekvenci fotonů a polarizaci. Na obou těchto veličinách totiž obecně závisí dělicí poměr děliče svazku a případné změny vlastností zdroje na těchto škálách by mohly vnést do generované náhodné sekvence nežádoucí korelace. V případě detektorů největší nebezpečí plyne z tzv. „afterpulsů", tj. u detektoru, který zaznamenal detekci, se zvyšuje pravděpodobnost, že zaznamená „falešnou detekci" (temný puls). Toto nebezpečí je minimalizováno na zanedbatelně malou úroveň pomocí detekční elektroniky.

1.2.3 Kontrola procesu dělení pulsu

Dalším faktorem, který je třeba experimentálně kontrolovat, je rovnovážnost generátoru, tj. zajištění toho, aby počet generovaných nul a jedniček byl s vysokou přesností stejný. Není totiž triviální zajistit, aby dělicí poměr děliče (spolu s detekční účinností detektorů) byly stabilně nastaveny na stejnou pravděpodobnost detekce na obou detektorech s přesností výrazně lepší než 1 %.

Jednou z možností jak tento nedostatek zmírnit je použít techniku tzv. **XORování**. Touto metodou vytvoříme jeden náhodný bit vždy ze dvou po sobě následujících úspěšných (detekce buď na detektoru A nebo na detektoru B) realizací experimentu podle následujícího klíče:

2. realizace	*1. realizace*	*výsledný bit*
A	A	0
B	A	1
A	B	1
B	B	0

Pak pravděpodobnosti generace bitů 0 a 1 jsou ($p_A = 1 - p_B$):

$$p_0 = p_A p_A + p_B p_B = 1 - 2 p_A (1 - p_A),$$
$$p_1 = p_A p_B + p_B p_A = 2 p_A (1 - p_A).$$

Pokud např. $|p_A - 1/2| = 1$ %, pak $|p_0 - 1/2| = 0{,}04$ %. Tohoto zlepšení se ale dosahuje na úkor snížení rychlosti generace na jednu polovinu.

Pro uvažované kryptologické aplikace by však ani takové zlepšení nemuselo být dostatečné. Proto využíváme následující **metodu vyvážení generátoru, pocházející od von Neumanna.** Opět je vytvořen jeden logický náhodný bit vždy ze dvou po sobě následujících

úspěšných realizací experimentu. Dvojice AA a BB však jsou vyřazeny a dvojice AB a BA použity podle následujícího klíče.

2. realizace	*1. realizace*	*výsledný bit*
A	B	0
B	A	1

Při této metodě se sice rychlost generace náhodných bitů sníží v průměru na jednu čtvrtinu, ale platí $p_0 = p_1 = p_A p_B$ (pro libovolné = $p_A p_B$).

Aby se eliminoval vliv potenciálních dlouhodobých pomalých (např. teplotních) změn detekčních pravděpodobností p_A a p_B, započítávají se dvojice AB, resp. BA, pouze tehdy, leží-li odpovídající detekce uvnitř zvoleného časového intervalu.

1.2.4 Detekce

Detektory pro detekci jednotlivých fotonů nemají ideální (100%) detekční účinnost, navíc detekční účinnosti u dvou detektorů použitých na výstupech děliče svazku nemají v praktickém systému stejnou hodnotu. Pro tuto aplikaci lze však v konkrétním systému zahrnout hodnoty detekčních účinností do dělicího poměru, tj. vyvážit nerovnovážnost detekčních účinností nastavením dělicího poměru děliče svazku tak, aby pravděpodobnost detekce na obou detektorech dosahovala potřebné hodnoty (zpravidla požadujeme stejnou pravděpodobnost detekce na obou detektorech).

Detektory vykazují dva druhy „falešných detekcí“, tj. poskytují elektronický puls i v případě, že na ně nedopadl žádný foton měřeného signálu:

a) Temné county: jednak to mohou být termální děje v lavinové fotodiodě, jednak šumové fotony vstupující do zařízení z jiných zdrojů než signálový laser. Množství termálních countů se omezuje termoelektrickým chlazením a volbou dostatečně krátkého detekčního časového okna (námi používané detektory vykazují 30-100 temných countů za sekundu, což při detekčním okně 10 ns širokém dává pravděpodobnost zachycení temného countu $<10^{-6}$).
b) Afterpulsy: po dopadu pulsu na detektor a detekci fotonu se zvyšuje pravděpodobnost vyslání falešného pulsu v důsledku nedokonalého návratu detektoru do základního stavu. Toto nebezpečí se eliminuje dostatečně dlouhou prodlevou mezi po sobě následujícími detekcemi. (V našem případě je opakovací frekvence omezena na 100kHz jinými faktory, což činí výskyt afterpulsů zanedbatelně malým.)

1.2.5 Generace dat

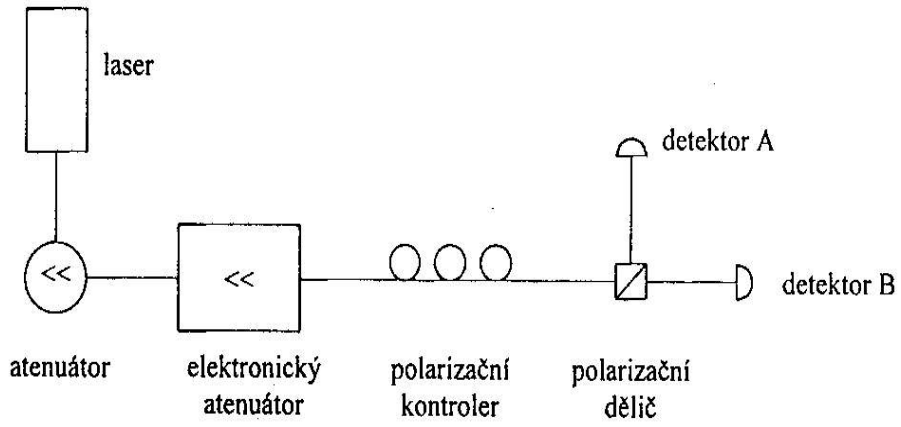
Technické možnosti, tj. maximální opakovací frekvence laserových pulsů, rychlost detekce (daná mrtvou dobou detektorů) a rychlost návazné detekční elektroniky, omezují maximální rychlost generace náhodných bitů. Největší omezení je na straně detektorů (doba nutná na uhašení lavinového procesu a vyčištění PN-přechodu) a především v detekční elektronice (zpracování obvody TAC a SCA a transfer dat do PC). Naše zařízení může pracovat s opakovací frekvencí laseru kolem 100 kHz. Pouze asi polovina pulsů bude správně detekována (nepoužijí se případy, kdy došlo k detekci na obou detektorech nebo na žádném z nich). Z úspěšných detekcí bude dál zužitkována asi čtvrtina (viz předchozí výklad). Lze tedy očekávat něco kolem 10 000 náhodných bitů za sekundu. Skutečné hodnoty v provedeném experimentu jsou 11 500 bitů za sekundu, tj. přibližně 5 megabyte za hodinu.

Vzhledem k tomu, že chyba odchylky průměru od ½, tedy veličiny $\left|\frac{1}{2} - \frac{\sum x}{n}\right|$, činí pro binomické rozdělení $\frac{1}{2\sqrt{n}}$, je pro potvrzení rovnováhy nul a jedniček s přesností např. 10^{-5} nutné vygenerovat nejméně 10^{10} bitů náhodných dat. To při dané rychlosti zařízení (v naší laboratorní realizaci) zabere asi 10 dní. Omezení rychlosti však není fundamentální, týká se dané laboratorní realizace a použitých komponent.

Experimentální realizace

1.3 Experimentální uspořádání

Schéma optické části experimentálního uspořádání je načrtnuto na následujícím obrázku. Zdrojem pulsů je polovodičový laser pracující na vlnové délce 830 nm, který je schopen generovat pulsy 400 ps až 4 ns dlouhé s opakovací frekvencí 100 Hz až 1 MHz. Každý puls obsahuje řádově 10^8 fotonů. Pulsy jsou zeslabeny nejprve mechanickým atenuátorem a poté je střední počet fotonů nastaven elektronických atenuátorem tak, aby součet intenzit na obou detektorech odpovídal vstupní intenzitě před děličem svazku 1,38 fotonu na puls. Dělič svazku s měnitelným dělicím poměrem je zkonstruován pomocí dvojice prvků – polarizačního kontroleru a polarizačního děliče svazku. Nastavením polarizačního stavu na vstupu polarizačního děliče lze dosáhnout libovolného dělicího poměru. Výstupy děliče svazku jsou sledovány detektory založenými na lavinových fotodiodách s kvantovou účinností okolo 50% na 830 nm.



Signály z detektorů jsou zpracovány pomocí detekční elektroniky sestávající z převodníků čas- amplituda a z jednokanálových analyzátorů, jejichž výstupy jsou sledovány z řídícího PC, které rovněž zajišťuje řízení elektronického atenuátoru, spouštění laseru a synchronizaci celého zařízení.

1.4 Experimentální výsledky

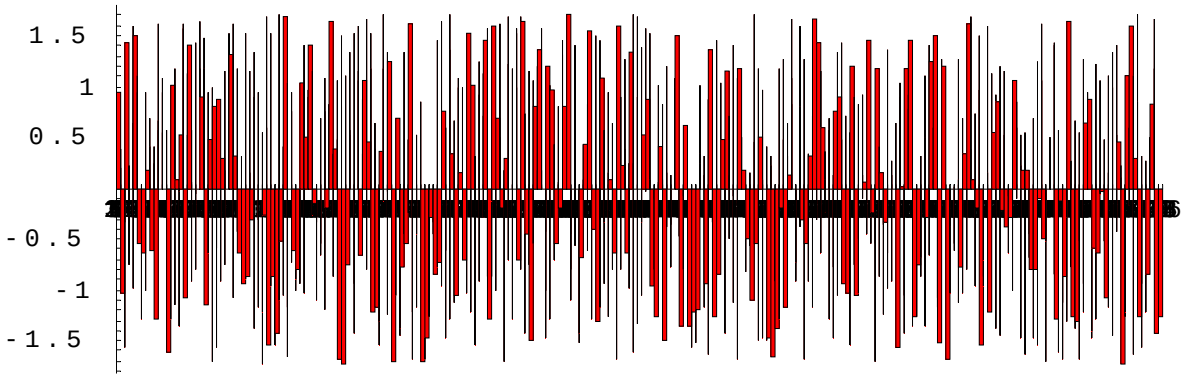
Na zařízení zkonstruovaném podle výše popsaného schématu byla provedena generace náhodných posloupností bitů oběma způsoby popsanými v části 1.2, tj. metodou XORování i von neumannovskou metodou.

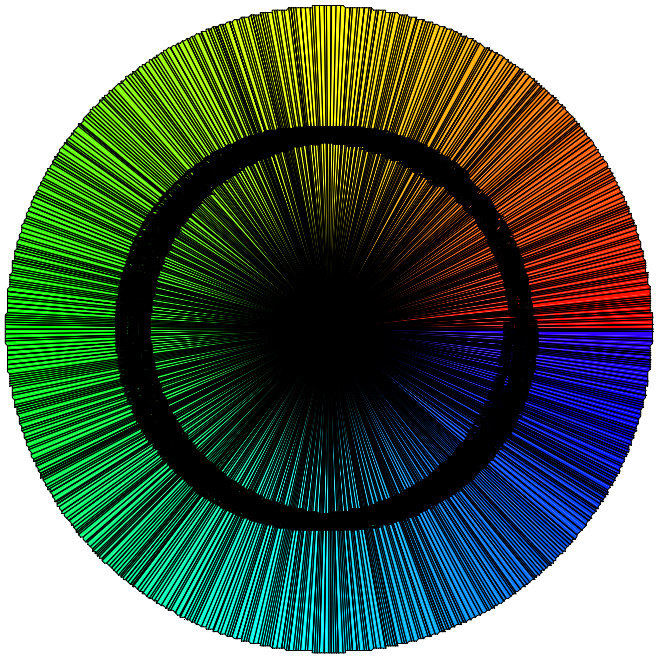
V případě generace pomocí XORování bylo dosaženo rychlosti generace 22,1 kbit/s. Dělicí poměr děliče svazku v průběhu měření kolísal v rozmezí 50,7:49,3 až 51,8:48,2 (typický průběh těchto změn lze vidět na následujícím obrázku) s průměrnou hodnotou 51,3:48,7. Metodou XORování by mělo dojít k vyvážení na hodnotu 50,03:49,97 (na vzorku 240 Mbit).

Von neumannovská metoda při podobném kolísání dělicího poměru teoreticky garantuje zcela vyvážený generátor.

Na vzorku dat získaných von Neumannovou metodou (výběr ze souborů Vn000-Vn073 na CD) byla zjištěna odchylka průměru od ½ o velikosti $4{,}8.10^{-5}$. Tato odchylka je menší než předpokládaná statistická fluktuace průměru ($8{,}6.10^{-5}$).

Statistický rozbor dat byl proveden programem Mathemathica, kde všechny soubory prokazovaly chaos, tj. kvasináhodný proces, jak je patrno z grafického znázornění





Jednotlivé soubory rovněž nejsou nikterak korelovány.

1.5 Závěr hodnocení dat z kvantového šumátoru

Zařízení ve Společné laboratoři optiky UP a FZÚ AV ČR funguje stabilně a je schopno generovat kvalitní náhodná data vhodná pro kryptografické aplikace. Dařilo se dobře kontrolovat všechny klíčové parametry a omezovat vnější rušivé vlivy. Posuzováno podle průběžně prováděných testů rovnováhy nul a jedniček, jeví generovaná data poměrně dobrou kvalitu (odchylky leží v rozsahu statistických chyb). K serióznímu posouzení je nutné srovnání s ostatními fyzikálními generátory náhodných čísel.

Využití jiného fyzikálního hardwaru ke generaci náhodných čísel

V oblasti fyzikálního hardwaru objevujeme kvalitní možnosti pro tvorbu přenositelných systémů generujících opravdu náhodné hodnoty pro kryptografické aplikace. Vše co k tomu potřebujeme je fyzikální zdroj náhodných (neuhodnutelných) hodnot.

Takovýmito zdroji může být například tepelný šum, radioaktivní rozklad nebo dva rychle vázané kmitající oscilátory. Jediným problémem pak je, jak hodnoty získané pomocí těchto fyzikálních jevů správně navzorkovat a zpřístupnit daným algoritmům k využití.

Objem požadovaných dat

Kolik náhodných dat vlastně potřebujeme? Dá se specifikovat množství těchto požadavků například počtem náhodných bitů za sekundu?

Například u algoritmu DES potřebujeme pro vygenerování 56-ti bitového klíče, při zajištění nejvyšší bezpečnosti pole 200 náhodných bitů. K tomu můžeme využít kryptograficky silných sekvencí tak jak jsou popsány v kapitole 1.9. Pokud potřebujeme pouze několik stovek náhodných bitů denně a máme k dispozici pomalý generátor s možností generovat např. jeden bit za sekundu, tak lze v plné míře tolerovat, v rámci udržení vysoké míry bezpečnosti, například dvousetsekundové zastavení (čekání) takové aplikace jednou za den.

1.6 Míra nesouměrnosti

Má mít vygenerovaná posloupnost náhodných hodnot určité specifické rozložení? Dobrou zprávou je, že rozložení takovéto posloupnosti nemusí být uniformní. V následujících kapitolách jsou popsány jednoduché principy, jak kontrolovat nesouměrnost takovýchto posloupností bitů.

1.6.1 Použití proudové parity

Vezmeme v úvahu dostatečně dlouhý bitový řetězec skládající se z určitého počtu nul a jedniček. Zobrazením tohoto řetězce nedostaneme přesně uniformní rozložení, ale můžeme si zvolit požadovanou odchylku od tohoto rozložení, které se budeme držet. K určení této míry nesouměrnosti (přesněji řečeno nevyváženosti počtu nul a jedniček) nám pomůže výpočet parity. Tuto proudovou paritu nám může počítat jednoduché HW zařízení.

Pomocí následujícího rozboru získáme vzorec pro určení délky bitových vzorků, pomocí jejichž parity můžeme tuto nesouměrnost kontrolovat.

Předpokládejme poměr výskytu jedniček k nulám 0,5+e : 0,5-e, kde *e* vyjadřuje výstřednost rozložení. Uvažujme výpočet paritní funkce pro n-bitové vzorky. Pravděpodobnost, že výsledná parita bude jedničková nebo nulová je dána sumou lichých a sudých vzorků v binomickém rozšíření $(p+q)^N$, kde p=0,5+e (pravděpodobnost výskytu jedniček) a q=0,5-e (pravděpodobnost výskytu nul).
Tuto sumu můžeme vypočíst podle následujících vztahů:

$$\frac{1}{2}*((p+q)^N+(p-q)^N) \text{ a } \frac{1}{2}*((p+q)^N-(p-q)^N)$$

kde jeden ze vztahů je pro lichý a druhý pro sudý počet vzorků.

Vezmeme-li v úvahu, že *p+q=1* a *p-q=2e*, můžeme tyto výrazy zjednodušit na:

$$\frac{1}{2} * (1 + (2e)^N) \text{ a } \frac{1}{2} * (1 - (2e)^N)$$

Z těchto vztahů je vidět, že pokud má být tato pravděpodobnost vyvážena, musí být *e* rovno 0. My si v této fázi můžeme zvolit libovolně velké okolí hodnoty pravděpodobnosti 0,5 a označit je jako *d*. Tato hodnota nám bude zaručovat určitý maximální stupeň nesouměrnosti a můžeme podle ní vypočíst délku bitových vzorku (N) pro výpočet parity.

$(0.5 + (0.5 * (2e)^N)) < 0.5 + d$, z čehož pro N plyne: $N > \frac{\log(2d)}{\log(2e)}$

Následující tabulka nám ukáže délku bitových vzorků, pro měření parity, pro dané stupně nesouměrnosti. Použitá odchylka *d* je 0,001.

pravděpodobnost(1)	E	N
0,5	0,0	1
0,6	0,1	4
0,7	0,2	7
0,8	0,3	13
0,9	0,4	28
0,95	0,45	59
0,99	0,49	308

Tabulka 0-1 – délka bit. vzorků pro dané stupně nesouměrnosti

Poslední řádek tabulky nám ukazuje, jak dlouhý má být vzorek pro měření parity, pokud bude vyžadována 99% nesouměrnost ve prospěch jedniček.

1.6.2 Párová nesouměrnost

Další technika spočívá ve zkoumání bitové sekvence jako množiny nestřídajících se bitových párů. Předpokládaná pravděpodobnost výskytu jedniček je opět *0,5+e* a nul *0,5-e*, kde *e* je výstřednost stejně tak jako v minulém algoritmu. Pravděpodobnost výskytu takovýchto párů definuje následující tabulka.

pár	Pravděpodobnost	
00	$(0,5-e)^2$	$0,25 - e + e^2$
01	$(0,5-e)* (0,5+e)$	$0,25 - e^2$
10	$(0,5+e)* (0,5-e)$	$0,25 - e^2$
00	$(0,5+e)^2$	$0,25 + e + e^2$

Tabulka 0-2 – pravděpodobnost výskytu bitových párů

Tato technika předpokládá vstupní posloupnost takovou, kde pravděpodobnost výskytu jednotlivých hodnot je stejná jak pro jedničky tak i nuly a nevyskytují se zde žádné korelace (sladěnosti).

Tímto postupem můžeme odhalit klam, který by nám poskytovala standardní statistická analýza. Jestli bychom například testovali posloupnost kde by se po sobě jdoucí páry bitů pravidelně střídaly, standardní statistické testy by vykazovaly stejné výsledky jako kdyby se páry nestřídaly. Takto vytvořená posloupnost by pak byla lehce předpověditelná.

1.6.3 Rychlá Fourierova transformace

Pokud se reálná data skládají ze silně zkreslených a korelovaných (majících určité zákonitosti) posloupností bitů, mohou nám stále poskytovat dostatečnou míru náhodnosti. Tato náhodnost může být ze vstupních dat „vytažena“ použitím diskrétní Fourierovy transformace (FT) nebo její modifikované metody rychlé Fourierovy transformace (FFT).

Použitím FT jsou silné korelace z posloupnosti odstraněny a výsledné spektrum je možno považovat za dostatečně náhodné.

1.6.4 Kompresní techniky

Neztrátové kompresní techniky také poskytují surovou metodu pro korekci náhodných sekvencí do ještě méně předpověditelného tvaru.Pokud použijeme zpětně neztrátovou kompresní metodu, pak musí být, podle Shanonovy věty, obsažena v krátké výstupní posloupnosti stejná míra informace jako je ve dlouhé vstupní posloupnosti. Použitím takovéto kompresní metody dostáváme více uniformně rozloženou posloupnost o což nám přesně jde.

Bohužel, ale mnoho kompresních technik přidává do výstupní posloupnosti určité předpověditelné fráze. Algoritmus pak sice zbaví původní posloupnost opakujících se frází, ale zároveň přidá do posloupnosti nové fráze a to své vlastní. V takovémto případě je dobré alespoň se vyvarovat použití několika počátečních bitů výsledné posloupnosti. Poznamenejme, že v poslední době se úspěšně rozvíjejí kompresní techniky na báze nelineárních metod, konkrétně pak tzv. fraktálních transformací.

Nehardwarové metody

Jaká je nejlepší strategie vyhovění požadavku „neuhodnutelnosti“ náhodných čísel při absenci spolehlivého HW zdroje anebo pro vylepšení kvality dat hardwarového zdroje. Jednou z možných metod je získat náhodná data z velkého počtu nezávislých zdrojů a sloučit je dohromady za pomoci některé kvalitní směšovací funkce. Pokud bude tato funkce střídat zdroje podle pevného nebo snadno uhodnutelného předpisu opět ztratí možnost tvořit dostatečně náhodnou posloupnost. Způsob pevného střídání určitých zdrojů je použitelný v případě některých často chybujících HW zařízeních, pokud se chceme vyhnout softwarovému řešení.

1.7 Směšovací funkce

Silnou směšovací funkcí myslíme takovou funkci, která z dvou nebo více vstupních proudů produkuje takovou výstupní posloupnost, kde každý výstupní bit je dán jinou složenou nelineární funkcí všech bitů vstupních. Obecně vzato změna jediného vstupního bitu by měla vyvolat změnu přibližně poloviny bitů výstupních. Protože vstupně výstupní vztah je komplexní a nelineární, žádný jednotlivý výstupní bit nemá zaručenu změnu své hodnoty, při změně některého konkrétního bitu vstupního.

Uvažujme problém konverze vstupního proudu bitů, na kratší posloupnost bitů výstupních, pomocí určité kompresní funkce. Toto je jedna z dalších možností jak navrhnout silnou směšovací funkci. Dalšími možnostmi, jak navrhnout vhodnou směšovací funkci se budeme zabývat v následujících kapitolách.

1.7.1 Jednoduché funkce

Nejjednodušším příkladem „míchací“ funkce dvou vstupních posloupností bitů může být nonekvivalence (XOR), provedená mezi jednotlivými vstupními bity, tak jak je ukázána v následující tabulce. Je to sice nevhodný případ, kde změna jednoho ze vstupních bitů vyvolá vždy změnu patřičného výstupního bitu, ale jednoduchost tohoto příkladu nám poskytuje užitečnou ilustraci.

1. vstup	2. vstup	Výstup
0	0	0
0	1	1
1	0	1
1	1	0

Tabulka 0-3 – funkce nonekvivalence (XOR)

Jestliže mezi vstupními posloupnostmi neexistuje žádná známa funkční závislost, pak výstupní sekvence bude mít lepší vlastnosti, než sekvence vstupní (menší nesouměrnost). Jestliže bychom chtěli vypočíst výstřednost, tak jak je definována v kapitole 1.6., výstupní posloupnosti, na základě znalosti výstředností vstupních posloupností, mohli bychom to provést pomocí následujícího vztahu:

$e = 2 * e_1 * e_2$, kde e_1 a e_2 jsou výstřednosti původních vstupních posloupností.

Protože e není nikdy větší jak *0,5,* tak hodnota výsledné výstřednosti bude vždy o něco lepší než jakákoliv hodnota výstřednosti vstupních posloupností (pouze v případě, kdy jedna vstupní posloupnost budě tvořena posloupností např. samých jedniček, pak bude hodnota výstřednosti výstupní posloupnosti shodná s hodnotou výstřednosti druhé vstupní posloupnosti).

Následující tabulka nám ukazuje hodnoty několika možných vypočtených výstředností, pro několik vstupních hodnot výstředností.

e	0,00	0,10	0,20	0,30	0,40	0,50
0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,10	0,00	0,02	0,04	0,06	0,08	0,10
0,20	0,00	0,04	0,08	0,12	0,16	0,20
0,30	0,00	0,06	0,12	0,18	0,24	0,30
0,40	0,00	0,08	0,16	0,24	0,32	0,40
0,50	0,00	0,10	0,20	0,30	0,32	0,50

Tabulka 0-4 - příklady vypočtených výstředností

Mějme na paměti, že výše uvedené hodnoty jsou platné pouze v případě použití dvou na sobě nijak nezávislých vstupních zdrojů. Pokud bychom například jako vstupy použili dva různé, ale přesné časovače (hodiny), výsledná posloupnost by neměla nijak kvalitní vlastnosti, protože vstupní posloupnosti by na sobě byly dost silně závislé, tím že by byly brány „v podstatě“ ze stejného zdroje dat.

1.7.2 Kvalitnější směšovací funkce

Šifrovací algoritmus DES je příkladem velmi silné směšovací funkce. Jako vstup potřebuje 120 bitů (64 bitů jako data a 56 bitů jako klíč) a produkuje 64 výstupních bitů z nichž každý tento bit je dán nelineární funkcí všech bitů vstupních. Další silná šifrovaní funkce s podobnými vlastnostmi, může být použita jako zdroj vstupních bitů pro data a klíč.

Další dobrou skupinou kvalitních směšovacích funkcí jsou tzv. hašovací funkce jako SHS, MD2 MD4 a MD5. Tyto funkce berou libovolný počet vstupních bitů a produkují posloupnost výstupních bitů o dané délce (v případě SHS je to 160 bitů a v případě MD* je to 128 bitů).

Ačkoliv pouze hašovací funkce jsou navrženy pro rozdílné množství vstupních dat, DES a ostatní šifrovací algoritmy lze též použít pro rozdílná množství vstupních dat. Pokud potřebujeme více bitů než máme k dispozici, můžeme vstupní posloupnost doplnit o nuly, které zašifrujeme opět pomocí DESu. Jestliže potřebujeme větší množství výstupních dat než 64 bitů, můžeme použít kombinovaného míchání. Tak například můžeme vstupní posloupnost rozdělit na tři části A, B a C. Poté vždy jednu část zašifrujeme za použití zbývajících dvou částí jako šifrovacích klíčů, čímž dostaneme tři samostatné náhodné posloupnosti. Další možnost je reverzace klíčů a opakování celého postupu. Podobného postupu můžeme použít u hašovacích funkcí (rozdělení vstupní posloupnosti na několik částí pro které samostatně vypočteme jednotlivé haše). Ve všech těchto případech je ale třeba mít na paměti, že je nemožné získat vyšší míru náhodnosti, než která nám do samotných algoritmů vstupuje.

1.7.3 Diffie-Hellman

Diffie-Hellmanova výměna klíčů je technika, která zaručuje společné tajemství mezi dvěma subjekty, kde je výpočetně nemožné toto tajemství odhalit i když máme k dispozici všechny zprávy, které si mezi sebou tyto subjekty vyměnily. Toto tajemství je směsicí inicializačních hodnot vygenerovaných oběma subjekty. Jestliže uvažujeme tyto hodnoty jako náhodná čísla, tak výsledné sdílené tajemství v sobě zahrnuje náhodnost z obou těchto hodnot.

1.7.4 Rozšiřování vstupní posloupnosti

Všechny tzv. míchací funkce nám dávají, na svém výstupu, stejný nebo menší počet bitů než se jim dostává na vstupu. Žádná z těchto funkcí nemůže zvýšit počet nepředpověditelných bitů ve výstupní sekvenci. Například pokud budeme míchat čtyři 32-bitové vstupy, z nichž každý bude obsahovat pouze 12 neuhodnutelných bitů, získáme ve výsledku maximálně 48 nových neuhodnutelných bitů. Výstupní posloupnost sice můžeme libovolně roztáhnou na stovky či tisíce bitů, ale počet takto vzniklých sekvencí bude stejně pořád 2^{48}.

Je možné vypozorovat skutečnost, že smícháním (XOR) náhodného bitu s bitem konstantním získáváme opět bit náhodný. I když je toto pravdou, tak již není pravdou, že bychom tímto způsobem mohli rozšiřovat výstupní posloupnost. Například smíchání vždy jednoho náhodného bitu nejprve s jedničkou a po té s nulou nedostaneme nové dva náhodné bity protože výsledná dvojhodnota je vždy buď sekvence 01 nebo 10.

1.7.5 Další faktory ovlivňující výběr „míchací“ funkce

Hlavní výhodou použití algoritmu DES je skutečnost, že byl v minulosti podroben mnoha testům a nebyly v něm objeveny žádné nedostatky či chyby. Další výhodou je to, že je k němu dostupná rozsáhlá dokumentace a ve formě zdrojových textů je volně k dispozici ke stažení z mnoha anonymních FTP archívů. Co se týká algoritmů SHS a MD*, tak ty jsou sice o něco mladší a ne tak otestované, ale není důvod proč jim nevěřit. Mnohé jejich implementace jsou opět volně ke stažení z internetu.
Použití algoritmů DES, SHS, MD4 a MD5 není nijak licenčně omezeno. Co se týká volby mezi šifrovacími a hašovacími algoritmy je pravděpodobně vhodnější použití hašovacích algoritmů protože se na ně nevztahují podmínky vývozu šifrovacích algoritmů z USA.

1.8 Nehardwarové zdroje náhodnosti

Nejlepším zdrojem pro vstup směšovacích funkcí jsou HW zdroje náhodnosti jako např. přístupová doba k disku, zvukový vstup a radioaktivní rozpad. Pokud ale tyto možnosti nemůžeme využít máme k dispozici další možné zdroje jako je např. systémový čas, vstupně výstupní vyrovnávací paměti (buffery), uživatelská a HW sériová čísla a uživatelské vstupy. Bohužel tyto zdroje produkují za určitých okolností pouze omezenou míru nepředpověditelných hodnot.

Některé z těchto zdrojů mohou být dostatečnými zdroji náhodnosti a to především u víceuživatelských systémů, kde každý uživatel je potencionálním zdrojem náhodných hodnot. Pokud ale tuto metodu praktikujeme na nejvíce rozšířených osobních PC, může potencionální útočník odhadnout naši konfiguraci a snížit tak míru naší nepředpověditelnosti.

Použitím kvalitní míchací funkce můžeme překonat slabost jednotlivých vstupních sekvencí a tvořit tak přenositelné aplikace i na jednouživatelských systémech. V každém případě je nejvhodnější použití HW zdrojů.

V poslední době se jako míchací funkce zkoumá synchronizace dvou chaotických dynamických systémů, anebo dokonce synchronizace dvou dopředu speciálně naučených neuronových sítí. Těmito novými přístupy se zde ale detailně zabývat nebudeme. Doposud totiž nebylo rozhodnuto, jestli takové přístupy skýtají požadovanou bezpečnost.

1.9 Kryptograficky silné posloupnosti

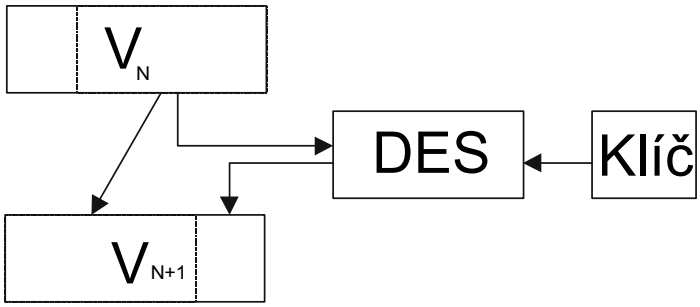
Obecně vzato útočník nesmí být schopen předpovědět žádnou hodnotu náhodného bitu a to ani v případě že ostatní hodnoty budou prozrazeny. Správným postupem při generování hodnot nějakým generátorem je zvolit silně náhodné startovací semínko a nikdy neodhalovat

celý stav generátoru ve výstupní posloupnosti, aby útočník nebyl schopen na základě znalosti předchozích bitů odhadnout hodnoty bitů následujících.

1.9.1 Tradiční silné posloupnosti

Jedna z možností získání opravdu silné kryptografické posloupnosti je využití pro generování některého šifrovacího algoritmu s náhodným klíčem a náhodnou startovací hodnotou. Na výstup takového generátoru je třeba zavést (pro některé nebo pro všechny výstupní bity) zpětnou vazbu a použít tyto bity v příští iteraci. Vhodná zpětná vazba pro daný šifrovací algoritmus je vždy jiná. Jedna možnost doporučená ve spojení s algoritmem DES je znázorněna na obrázku příkladu zpětnovazebního generátoru s využitím algoritmu DES.

Hodnota V_N představuje aktuální vnitřní stav generátoru a hodnota V_{N+1} představuje následující vnitřní stav. Nový stav je počítán ze starého využitím několika bitů v přímé vazbě a několika po zašifrování. Počet bitů v šifrované zpětné vazbě se pro algoritmus DES může pohybovat v rozmezí od 1 do plných 64 bitů. Stejný počet bitů lze také použít pro výstupní funkci v jedné iteraci.



Příklad zpětnovazebního generátoru s využitím algoritmu DES

Bylo dokázáno, že generující se posloupnost se může začít opakovat nejméně po 2^{64} vygenerovaných hodnotách při použití všech výstupních bitů a po 2^{31} až 2^{32} při použití jednoho až 63 výstupních bitů ve zpětné vazbě.

Předpovězení některého z generovaných bitů se rovná obtížnosti prolomení algoritmu DES s částečnou znalostí zašifrovaných hodnot. Čím menší počet vygenerovaných bitů použijeme ve výstupní posloupnosti, tím je možnost prolomení obtížnější. Nejbezpečnější tedy bude pokud v každém kole do výstupní posloupnosti přeneseme pouze jeden vygenerovaný bit.

1.9.2 Blum Blum Shubův generátor

Za nejodolnější současný softwarový generátor se považuje Blum Blum Shubův generátor, nazvaný podle po svých vynálezců. Jeho princip je velmi jednoduchý a je založen na výpočtu tzv. kvadratických zbytků. Jeho hlavní nevýhodou je bohužel vysoká výpočtová náročnost ve srovnání s generátory uvedenými v předchozí kapitole.

Na začátku si zvolíme dvě velká prvočísla, která obě mají následující vlastnost. Po vydělení těchto prvočísel číslem čtyři musí být výsledný zbytek roven třem. Označme si tato prvočísla jako *p* a *q*. Pak nechť *n=p*q*. Poté si zvolíme číslo *x* které je nesoudělné s *n*. Počáteční semínko generátoru a vzorec pro výpočet následujících hodnot je definován vztahem:

$$s_0 = \left(x^2\right)(\mathrm{mod}(n)) \text{ a } s_{i+1} = \left(s_i^{\,2}\right)(\mathrm{mod}(n))$$

Pro výstupní posloupnost musíme pečlivě vybírat pouze určitý počet nejnižších bitů těchto čísel. Nic nepokazíme, pokud jako výsledek vezmeme vždy pouze nejnižší bit čísla s_i. Jestli nepoužijeme více než $\log_2(\log_2(s_i))$ nejnižších bitů, tak předpovězení některého následujícího bitu na základě znalosti bitů předchozích je stejně těžké jako výpočet rozkladu čísla *n*. Pokud bude hodnota čísla *x* tajná, je možné zveřejnit hodnotu čísla *n*.

To znamená, že v aplikacích kde je použito velké množství takto vygenerovaných klíčů, nemusíme uchovávat celé tyto klíče, nýbrž pouze startovací hodnotu a modul *n*.

Zvláštní vlastností generátoru je to, že můžeme vypočíst jakoukoliv hodnotu čísla s_i ze znalosti následujícího vzorce:

$$s_i = \left(s_0^{\left((2^i)(\mathrm{mod}((p-1)(q-1)))\right)}\right)(\mathrm{mod}(n))$$

Více se tímto generátorem budeme zabývat ještě v pozdějších částech práce.

Testování vlastností generátorů

Po principech a doporučeních v předchozích kapitolách se zaměříme blíže na způsoby testování vlastností vygenerovaných posloupností. Je dobré si zde připomenout fakt, že pokud použitý generátor (ať už hardwarový či softwarový) nebude dostatečně kvalitní a dostane-li se do rukou odborníka, lze předpokládat, že tento odborník bude schopen odhalit veškeré potencionální chyby. Především bude schopen vidět konstrukční nebo algoritmické vady a dobu nebo okolnosti, kdy se mohou projevit.

V případě sériové výroby příslušného hardwaru nebo nasazení stejného softwaru v mnoha zařízeních si pak potencionální útočník může pro svůj útok vybrat nejvhodnější místo. Proti hardwaru někdy pomůže magnet, jindy zvýšení teploty, nebo naopak trocha tekutého dusíku pro ochlazení přístroje, a to jen proto aby se generátor „omámil“, „uspal“ nebo dostal do nedefinovaného, chybového nebo jiného vytouženého stavu. Zkrátka je třeba počítat s tím, že lidé kteří se tímto druhem „živnosti“ zabývají, mají dostatečně bujnou fantazii i motivaci.

Připomeneme-li si i útoky na prolomení algoritmu DES hrubou silou a vyzkoušení 35*10^18 jeho klíčů „jen tak pro radost“, je zřejmé, že při návrhu generátorů náhodných čísel nebo posuzování jeho kvality v jakémkoliv bezpečnostním systému nemůže být shovívavost na místě. Proto také vznikly normy, které definují minimální požadavky na kvalitu těchto generátorů.

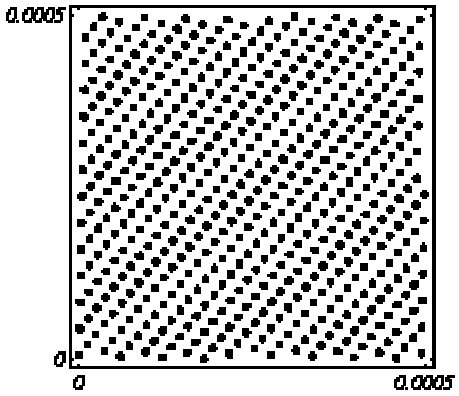
Těmto normám se budeme podrobněji věnovat v části 1.12. Následující dvě podkapitoly se budou věnovat základním teoretickým a empirickým testům.

1.10 Teoretické testy

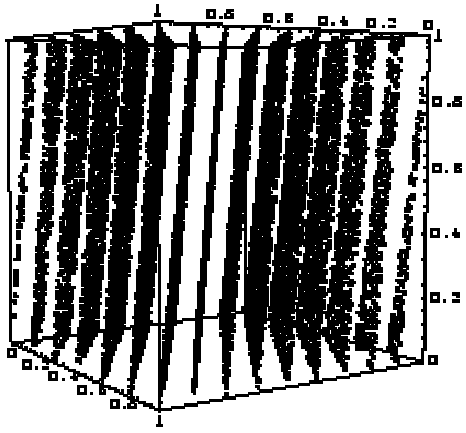
Do této kategorie spadají zejména metodiky návrhu generujících algoritmů, tak aby co nejlépe splňoval požadavky definované v předchozích kapitolách. Teoretické testy se nezaměřují ani tak na testování vygenerovaných posloupností, jako spíše na zkoumání vlivu jednotlivých parametrů generátoru na výslednou vygenerovanou posloupnost.

Do této kategorie spadá například pozorování vygenerované posloupnosti ve formě několika-bitových čísel. Pouhým okem je pak možné v grafickém provedení vypozorovat případné korelace (shody) v náhodné posloupnosti, které by neodhalil žádný z testů. Dalším testem může být i pozorování četnosti vygenerované posloupnosti, která by měla být téměř pro všechny vzorky (několika-bitová čísla) obdobná.

Nejlepší teoretické testy jsou založené na pozorování náhodných čísel vsazených do virtuálních několika-dimenzionálních prostorů. To znamená, že pokud máme posloupnost U1, U2, … náhodných čísel, pak tyto čísla přemístíme do k-rozměrných jednotkových krychlí tak, že vždy posloupnost (U1, U2, … , Uk) bude tvořit jeden bod v k-rozměrném souřadném systému. Pro příklad si uvedeme k rovno dvěma, pak budou uspořádané dvojice (U1, U2), (U3, U4), .. tvořit jednotlivé body na ploše. Takovýmto útvarů se pak říká mřížkové plochy.



Př.3- ANSI LCG (2^31, 1103515245, 12345, 12345)



Př.2- RANDU LCG (2^31, 2^16+3=65539, 0, 1)

V dnešní době se nejčastěji využívá dvojrozměrných a třírozměrných křížových ploch. Čím jsou výsledné plochy hustěji a náhodněji osázeny tím je výsledný generátor kvalitnější. U pseudo- náhodných generátorů nejsou body v mřížových plochách rozmístěny náhodně všude, ale jen na určitém počtu nadrovin (nadrovinou zde myslíme prostor o jednu dimenzi nižší než prostor výchozí). Na obrázcích si můžeme prohlédnout ukázky mřížových ploch pro některé skutečné pseudonáhodné generátory.

Mezi další možnosti teoretických testů může například patřit komprese vygenerované bitové posloupnosti. Pokud je posloupnost kvalitní, neměli by se v ní vyskytovat žádné korelace, kterých kompresní techniky využívají. Pokud tedy po „zabalení“ vygenerované posloupnosti bude mít soubor tutéž velikost, je generátor pravděpodobně bez korelací a tudíž i dostatečně náhodný.

1.11 Empirické (statistické) testy

Pro testování náhodných čísel nejprve zavedeme univerzální statistické kritérium χ^2, kterého využíváme téměř ve všech testech tímto způsobem:

Všechna náhodná čísla rozdělíme do k kategorií a provádíme *n* navzájem nezávislých pokusů, to znamená, že n-krát generujeme veličinu s rovnoměrným nebo jiným pravděpodobnostním rozložením. Symbolem p_s označíme pravděpodobnost, že výsledek pokusu padne do kategorie *s* a y_s nechť je počet pokusů, které skutečně padly do kategorie *s*. Poté můžeme zformulovat vztah

$$V = \sum_{1 \le s \le k} \frac{(y_s - n \cdot p_s)^2}{n \cdot p_s} \qquad \text{– vzorec (CHI)}$$

kde *V* je výsledná hodnota testu χ^2.

	alpha ==>									
ν	0.995	0.99	0.975	0.95	0.9	0.1	0.05	0.025	0.01	0.005
1	0.000	0.000	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.832	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.647	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.041	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401
22	8.643	9.542	10.982	12.338	14.041	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.558
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.878	14.573	16.151	18.114	36.741	40.113	43.195	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.994
29	13.121	14.256	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.335
30	13.787	14.953	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
31	14.458	15.655	17.539	19.281	21.434	41.422	44.985	48.232	52.191	55.002
32	15.134	16.362	18.291	20.072	22.271	42.585	46.194	49.480	53.486	56.328
33	15.815	17.073	19.047	20.867	23.110	43.745	47.400	50.725	54.775	57.648
34	16.501	17.789	19.806	21.664	23.952	44.903	48.602	51.966	56.061	58.964
35	17.192	18.509	20.569	22.465	24.797	46.059	49.802	53.203	57.342	60.275
36	17.887	19.233	21.336	23.269	25.643	47.212	50.998	54.437	58.619	61.581
37	18.586	19.960	22.106	24.075	26.492	48.363	52.192	55.668	59.893	62.883
38	19.289	20.691	22.878	24.884	27.343	49.513	53.384	56.895	61.162	64.181
39	19.996	21.426	23.654	25.695	28.196	50.660	54.572	58.120	62.428	65.475
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.170
150	109.142	112.668	117.985	122.692	128.275	172.581	179.581	185.800	193.207	198.360
200	152.241	156.432	162.728	168.279	174.835	226.021	233.994	241.058	249.445	255.264

Tabulka 0-5 - kritické hodnoty pro χ^2 rozložení

Když jsme získali hodnotu *V*, musíme určit, zda je tato hodnota vyhovující. Zavádíme proto stupeň volnosti, jenž je v našem případě roven *v=k-1*, to znamená že stupeň volnosti je o jedničku menší, než je počet kategorií.

Známe-li výsledek testu χ^2 a stupeň volnosti, můžeme podle statistických tabulek určit hladinu významnosti, které tato hodnota odpovídá (příkladem statistické tabulky je tabulka 0.1, kde *v* je stupeň volnosti a α je odpovídající hladina významnosti pro vypočtenou hodnotu *V*).

Hladina významnosti je předem zvolená pravděpodobnost, která je dostatečně malá pro zamítnutí hypotézy o rozložení daného náhodného čísla. Hladinu významnosti značíme symbolem α. Při využití testu χ^2 postupujeme zpravidla takto: nejdříve zjistíme hodnotu *V*, pro tuto hodnotu a pro příslušný stupeň volnosti najdeme odpovídající hladinu významnosti, případně její nejbližší nižší hodnotu.

Generátor náhodných čísel pokládáme za vyhovující, je-li hladina významnosti větší než 0,90. Samozřejmě musíme dbát na to, aby byl počet pokusů *n* co největší. V případě, že je teoretický počet čísel, které padnou do s-té kategorie, menší než 5, nezahrnujeme tuto kategorii do výpočtů a stupeň volnosti *v* je tedy nižší.

1.11.1 Testy rovnoměrnosti rozložení

V těchto testech využíváme vzorce (CHI), podle něhož přímo srovnáváme skutečnost s teoretickými hodnotami.

Test na rovnoměrnost

Interval, do něhož padnou generované náhodné veličiny, se rozdělí do *n* podintervalů; pravděpodobnost, že generovaná veličina padne obecně do k-tého intervalu je 1/n. Pak statisticky zhodnotíme kritériem χ^2 teoretický počet generovaných veličin, které padnou do jednotlivých intervalů a jejich skutečný počet. Můžeme také brát v úvahu dvojice až n-tice sousedních čísel a zjišťovat, kolik jich padne do dvou až n-rozměrných intervalů.

Test dvojic

Jednotkový čtverec rozdělíme na větší počet (asi 100) dílů. Potom bereme po sobě následující dvojice náhodných čísel a zjišťujeme, kolik jich padne do jednotlivých intervalů (dvojici považujeme za souřadnice bodu, který padne do jednotkového čtverce). V jedné sérii zkoumáme asi 50 tisíc hodnot. Zhodnocení výsledků metody provede opět testem χ^2, přičemž jako stupeň volnosti budeme volit $(n-1)^2$, kde *n* je rozměr strany čtverce. Celý čtverec tedy obsahuje *n.n* položek.

Rovnoměrnost trojic

Při tomto testu postupujeme stejně jako v předcházejícím, bereme však trojice náhodných čísel a uvažujeme jednotkovou krychli. Volíme asi 1000 intervalů a 30 tisíc náhodných hodnot.

Rozložení maxima z *n* členů

V tomto testu bereme n-tice po sobě následujících náhodných čísel $x_1, x_2, x_3, \ldots, x_n$. Vypočítáme hodnoty $U=[max(x_1, x_2, x_3, .., x_n)]^n$. Takto vzniklou posloupnost hodnot *U* testujeme na rovnoměrnost. Volíme $n=2, 3, 4, 5, 10$ a v jednom pokusu testujeme 100 tisíc hodnot.

1.11.2 Testy náhodnosti rozložení

Rozhodnutí o náhodnosti posloupnosti vygenerovaných není jednoduché, protože v konečné posloupnosti náhodných čísel můžeme vždy nějakou zákonitost najít. Proto zjišťujeme, do jaké míry je tato posloupnost náhodná a tuto míru potom kvantitativně ohodnotíme.

Test na intervaly

V tomto testu prověřujeme délku posloupnosti náhodných čísel mezi dvěma hodnotami, které padnou do téhož předem zvoleného intervalu. Vyčíslíme tedy počty posloupností stejných délek a statisticky je zhodnotíme kritériem χ^2.
Zavedeme-li hodnotu *p* = *horní mez intervalu – dolní mez* intervalu, pak pravděpodobnost, že posloupnost bude mít délku *t*, je: $p_t=p(1-p)^t$, tedy pro posloupnost délky nula platí $p_0=p$. Tohoto testu se dá využít pouze pro generátor čísel typu *real* s rozložením R<0,1>.
Stupeň volnosti χ^2 pro tento test volíme *e / (horní mez intervalu–dolní mez)*, přičemž *e* je základ přirozeného logaritmu. Toto tvrzení plyne ze skutečnosti, že pokud *p* je šířka intervalu, pak *p* je taky pravděpodobnost že hned další číslo padne do téhož intervalu. Nejsnažší cestou je, že *1/p*-té číslo padne do téhož intervalu, pravděpodobnost tohoto tvrzení je rovna $p_t=p(1-p)^{(1/p)}$, přičemž *t = 1/p.* Výraz $(1-p)^{(1/p)}$ *se* blíží hodnotě *1/e*. Z tohoto tvrzení plyne, že výsledek výrazu p_t se vždy blíží hodnotě *p/e*. A odtud *df* (stupeň volnosti) = *e/p*.

Test sběratele kupónů

Zjišťuje se délka posloupnosti náhodných čísel taková, že se v ní každý možný člen posloupnosti objeví alespoň jednou. Pro kritérium χ^2 se používá vztahu $q_r = 1 - \frac{d!}{d^r}\{{}^{r}_{d}\}$, kde {r nad d} je Stirlingovo číslo druhého druhu, q_r je pravděpodobnost, že posloupnost délky *r* neobsahuje všechna čísla generovaná generátorem, *d* je největší možné číslo, které generátor může vytvořit. Test sběratele kupónů se dá použít pro generátor čísel typu integer v intervalu <0, d-1>. Stirlingova čísla se zapisují ve tvaru: $x=\{{}^{a}_{k}\}$.

STIRLING NUMBERS S(p, k) OF THE SECOND KIND

	k=	0	1	2	3	4	5	6	7	8	9	10
p=												
0		1										
1		0	1									
2		0	1	1								
3		0	1	3	1							
4		0	1	7	6	1						
5		0	1	15	25	10	1					
6		0	1	31	90	65	15	1				
7		0	1	63	301	350	140	21	1			
8		0	1	127	966	1701	1050	266	28	1		
9		0	1	255	3025	7770	6951	2646	462	36	1	
10		0	1	511	9330	34105	42525	22827	5880	750	45	1

Tabulka 0-6 – tabulka Stirlingových čísel druhého druhu

Test zvaný RunUp

V tomto testu sledujeme vždy neklesající sekvence čísel a počítáme četnosti jednotlivých délek těchto neklesajících posloupností. Četnosti počítáme pro délky jedna až pět a všechny delší zařazujeme do kolonky s délkou šest. Vezměme si pro příklad označení těchto četností r_i, kde *i = 0..6*. Výpočet χ^2 lze provést na základě vzorce

$$\chi^2 = \frac{1}{n}\sum_{i=1}^{6}\sum_{j=1}^{6} a_{ij}(r_i - n\cdot b_i)(r_j - n\cdot b_j)$$

kde *n* je počet vygenerovaných čísel, a_{ij} a b_i jsou následující konstantní matice:

a_{ij} = {{4529.4, 9044.9, 13568, 18091, 22615, 27892}, {9044.9, 18097, 27139, 36187, 45234, 55789}, {13568, 27139, 40721, 54281, 67852, 83685}, {18091, 36187, 54281, 72414, 90470, 111580}, {22615, 45234, 67852, 90470, 113262, 139476}, {27892, 55789, 83685, 11580, 139476, 172860}}

b_i = {1.0/6.0, 5.0/24.0, 11.0/120.0, 19.0/720.0, 29.0/5040.0, 1.0/840.0}

Počet stupňů volnosti je pro χ^2 test roven 6.

Test mezer

Uvažujeme-li tři sousední čísla posloupnosti, pak existuje právě 6 možností vzájemných relací: a<b<c, a>b>c, a<c<b, a>c>b, c<a<b, c>a>b. Možnost, že by se dvě nebo tři čísla v trojici rovnala, vylučujeme. Při vhodně zvolených konstantách generátoru se to nestává ani u pseudonáhodného generátoru. Zjistíme, kolikrát se ve zkoumané posloupnosti objeví každá z těchto možností a statisticky je zhodnotíme testem χ^2 s tím, že předpokládáme, že pravděpodobnost je vždy rovna jedné šestině a počet stupňů volnosti pro chi-kvadrát je 6.

Poker test

Test na rovnoměrnost by mohl dostat dobré výsledky i v případě posloupnosti čísel, která by byla nedostatečně náhodná. Příkladem takové posloupnosti je např. 11112222333344445555666…. Je zřejmé, že počet členů posloupnosti, které padnou do jednotlivých intervalů sice odpovídá rovnoměrnému rozložení, ale tato posloupnost se ani zdaleka nepodobá posloupnosti náhodné. Proto zavádíme poker test: Posloupnost generovaných čísel rozdělíme do pětic, pak zjistíme počet různých čísel v pětici, který označíme např. písmenem *r*. Pak použijeme následujícího kritéria s pravděpodobnostmi

$$p_r = \frac{d(d-1)..(d-r+1)}{d^k} \left\{ {k \atop r} \right\}$$

kde p_r je pravděpodobnost, že v pětici bude právě *r* různých čísel, *k* je počet čísel ve skupině (v našem případě 5, počítáme pětice), *d* je maximální možná hodnota generované veličiny zvětšené o 1, $\left\{ {k \atop r} \right\}$ je Strilingovo číslo druhého druhu.Tohoto testu se dá využít pouze pro testování posloupností, jejichž členy jsou čísla typu integer a padnou do intervalu <0,d-1>.

Hamingův test

Tento test již nespadá do třídy empirických testu, které využívají kritéria χ^2. Test má odhalit, zda se některé hodnoty náhodných čísel nevyskytují s větší četností. Zkoumá se velikost součtu:

$$\sum_{i=1}^{n-1} (y_i - 0.5)(y_{i+1} - 0.5) \to 0$$

kde *n* je počet vygenerovaných čísel.

Je-li generátor dobře navržen, je hodnota tohoto součtu přibližně nulová.

1.12 Kryptografické testy (normy)

Základním předpisem, který se zabývá mimo jiné kvalitativní mírou náhodnosti, je americká norma s názvem *Security requirements for cryptographic modules* zveřejněná ve *Federal Information Processing Standard Publication 140-1, U.S. Department of Commerce, National Institute for Standards and Technology, National Technical Information Service, 1994.* Je známá především pod zkratkou *FIPS PUB 140-1.*

Tato norma upřesňuje bezpečnostní požadavky pro návrh a implementaci kryptografických modulů pro ochranu (amerických vládních) neutajovaných informací, včetně hardwaru, softwaru, modulů a jejich kombinací. Jsou zde specifikovány čtyři úrovně bezpečnostních aplikací a zařízení, ve kterých je zakotveno kdy a které testy náhodnosti provádět. Standardizační úřad *NIST* také provádí testy, zda příslušné moduly splňují tyto normy.

První dvě úrovně jsou definovány pro čistě softwarové moduly a je u nich vyžadovaná menší míra bezpečnosti. Liší se počtem uživatelů přistupujících ke sdíleným prostředkům, druhá úroveň je uzpůsobena pro multi-uživatelské systémy. Třetí úroveň je definována jak pro softwarové tak i hardwarové moduly. Na této úrovni musí být v modulu zakotvena možnost provedení základních kryptografických testů náhodnosti na požádání. Nejpřísnější úroveň, která zhrnuje mimo jiné i odolnost proti vnějším vlivům prostředí jako je teplota či elektrické napětí, definuje provádění kryptografických testů náhodnosti jednak na požádání, ale také pravidelně po každém nastartování příslušného modulu (softwarového, hardwarového i kombinovaného).

Návrhář modulu musí zaručit, že modul vyhovuje všem předepsaným statistickým testům (kapitola 1.12.1), jinak dostává známku „nevyhověl“. Je-li schválený modul v provozu a (třeba následkem poruchy elektroniky) nevyhoví kontrolnímu testu, musí ohlásit chybový stav. Celá norma je dosti rozsáhlý dokument, který definuje další fyzikálně-bezpečnostní vlastnosti pro získání daných tříd bezpečnosti aplikací.

V následující kapitole si uvedeme základní statistické testy definované v normě FIPS PUB 140-1.

1.12.1 Testy náhodnosti podle normy FIPS PUB 140-1

Výsledky těchto testů jsou definovány pro vstupní posloupnost 20 000 náhodných bitů a mají přesně definovaný obor platných hodnot.

Monobit test – test četnosti jedniček

Modul vyhovuje tomuto testu, pokud ve vygenerované posloupnosti 20 000 bitů je počet jedniček v rozmezí od 9 654 do 10 346.

Poker test

Posloupnost 20 000 náhodných bitů je neřetězovitě rozdělena na 5000 čtyřbitových úseků, které reprezentují hodnotu *i = 0 .. 15*. Počet úseků s hodnotou i si označme jako *f(i)*. Podle následujícího vzorce pak vypočteme testovací hodnotu X

$$X = (\frac{16}{5000}) * (\sum_{i=0}^{15} [f(i)]^2) - 5000$$

Modul vyhoví tomuto testu pokud platí že 1,03 < X < 57,4.

Run test

Termínem *run* se označuje úsek dané posloupnosti, který je složen se samých nul (pak se nazývá *gap*) nebo jedniček (v tom případě se nazývá *blok*). Například v posloupnosti 00101111100000001 je na začátku gap délky 2, poté je blok délky 1, následuje gap délky 1, blok délky 5, gap délky 7 atd.. Při testu spočítáme v dané posloupnosti počet gapů a počet bloků délky 1, 2, 3, 4, 5, 6 a více. Všech dvanáct vypočítaných čísel musí ležet v následujících intervalech.

1	2 267 – 2 733
2	1 079 – 1 421
3	502 – 748
4	223 – 402
5	90 – 223
6 a více	90 – 223

Tabulka 0-7 – intervaly run testu

Test nejdelšího runu

V posloupnosti se sleduje, zda některý z gapů nebo bloků nedosáhl délky 34 nebo více, tj. zda posloupnost obsahuje posloupnost 34 nebo více nul za sebou nebo posloupnost 34 nebo více jedniček za sebou. Pokud ano, modul tomuto testu nevyhověl.

1.12.2 Následník FIPS PUB 140-1

Tato norma je v platnosti již od roku 1994 a v současné době se připravuje její modifikace, která mimo jiné zpřísňuje intervaly oboru výsledných hodnot testů. Její pracovní označení je FIPS PUB 140-2.

Monobit test – test četnosti jedniček

Modul vyhovuje tomuto testu, pokud je počet jedniček v rozmezí od 9 725 do 10 275.

Poker test

Modul vyhoví tomuto testu pokud platí, že hodnota X vypočtená podle rovnice 10.5. spadá do intervalu 1,03 < X < 57,4.

Run test

Délka runu	Interval
1	2 343 – 2 657
2	1 135 – 1 365
3	542 - 708
4	251 – 373
5	111 – 201
6 a více	111 - 201

Tabulka 0-8 – intervaly run testu pro FIPS PUB 140-2

Test nejdelšího runu

Nejdelší run může dosahovat velikosti 26 po sobě jdoucích stejných bitů.

1.13 Zhodnocení testů

V předchozích částech jsme si ukázali jak otestovat vygenerované náhodné posloupnosti a jak posuzovat jejich kvalitu. S ohledem na cílovou oblast použití realizovaných generátorů budeme klást největší důraz na kryptografické testy dané normou FIPS PUB 140-1.

Zařízení obsahující kvantový šumátor musí sloužit jako naprosto nepredikovatelný generátor náhodné posloupnosti nul a jedniček. Vlastní fyzikální proces, který stojí v pozadí tohoto generátoru popsaný v předchozích částech, má za předpokladu platnosti kvantové mechaniky náhodný charakter. Tato primární náhodnost je však dále zpracovávána zařízením postaveným z reálných součástek. Je všeobecně známo, že právě při následném

zpracování primární náhodné posloupnosti může docházet k nežádoucím jevům, které se promítnou do statistické kvality finálního produktu. Je proto nezbytně nutné produkci zařízení kvalitně statisticky testovat a vždy aplikovat matematické metody (např. XORování atd.) na syrová data získaná přímo z generátoru.

Tím se rozumí, že při převodu generované posloupnosti na výstupní posloupnost lze rovněž některé statistické parametry vylepšovat. U reálných true random generátorů se to týká zejména četnosti znaků "0" a "1". Je například známo, že u generátorů založených na šumu diod je maximálně dosažitelná rovnoměrnost výskytu nuly a jedničky okolo e @ 0.001, kde Pr(x="0")=0.5+e nebo Pr(x="0")=0.5-e. U kvantového šumátoru je výsledek o dva řády lepší.

Tyto charakteristiky lze zlepšit několika způsoby, z nichž nejpoužívanější je tzv. von Neumannova procedura viz [2].

Produkce kvantového generátoru byla dodána ve formě souborů o typické mohutnosti 1 gigabit. Při těchto velikostech lze testovat rovnoměrnost výskytu nuly a jedničky řádově e @ 0.0001.

Při konstrukci testovacího programu byly použity zdroje [3],[4]. V realizované verzi programu byly implementovány následující statistické testy:

1. POKER TESTY

Srovnává se výskyt všech možných n-tic bitů s jejich teoretickým výskytem.
Pro n=1 s normálním rozdělením, pro n>1 s příslušným chý-kvadrát rozdělením.
Podle velikosti souboru bylo testováno až pro n=15.
Poker testy velmi dobře odhalují případné odchylky od rovnoměrnosti výskytu jednotlivých znaků.

2. AUTOKORELAČNÍ TEST

Pro posuny 1 až 1024 byla spočtena četnost znaku "1" v posloupnosti vzniklé XORováním základní a posunuté posloupnosti. Výsledek byl standardizován do veličiny mající při platnosti hypotézy o nekorelovanosti normální rozdělení N(0,1).Takto získaných 1024 hodnot bylo srovnáno s normálním rozdělením N(0,1) pomocí Kolmogorov- Smirnovova nadtestu.Autokorelační testy odhalují případné časové závislosti generovaných znaků.

3. TEST SÉRIÍ

Sérií délky n rozumíme úsek délky n stejných binárních znaků na jehož obou koncích je znak opačný. Například ...100001... je série délky čtyři atd. Počty sérií délek 1 až 20 a delší než 20, byly porovnány s teoretickými počty pomocí chý-kvadrát testu. Pokud byl soubor kratší byla tomu přizpůsobena velikost maximální délky série. Tento test je využíván jako kritérium náhodnosti generované posloupnosti.

Pro celkové hodnocení každého souboru bylo provedeny všech jednotlivé. Za výsledek v každém testu se souboru udílely trestné body (TB):

- hypotéza přijata na hladině významnosti 0.05 .... 0 TB
- zamítnuta na 0.05 ale přijata na 0.01 .... 2 TB
- zamítnuta na 0.01 ale přijata na 0.001 .... 3 TB
- zamítnuta na 0.001 .... 6 TB

Na základě součtu trestných bodů je určeno celkové hodnocení souboru:

Známka:	TB:	Hodnocení:
1	0	VYHOVUJE
2	max 2	NEVYHOVUJE
3	max 5	NEVYHOVUJE
4	max 7	NEVYHOVUJE
5	8 a více	NEVYHOVUJE

Uvedenou metodikou bylo testovány vybrané soubory z množiny Vn000-073 a všechny po matematické úpravě dat vyhovovaly.

1.14 Nelineární metody

V případě tzv. impulsních generátorů náhodných čísel hrají roli náhodných momentů časové okamžiky (intervaly) realizace daného jevu. Takové systémy lze pak považovat za ergodické. Pro ty pak je popis pomocí amplitudy pravděpodobnosti ekvivalentní s popisem pomocí tzv. časových mapingů. Pro srovnání uveďme, že analogickým příkladem takového jevu je kódování v neuronálních systémech, a to jak na úrovni jednoho neuronu, tak i systému více neuronů uspořádaných do neuronové sítě. Z fyziky může jako příklad posloužit již zmiňovaný rozpad atomových jader v čase.

Pro systémy (generátory) tohoto typu a jejich testování lze pak použít metody nelineární dynamiky, resp. tzv. chaodynamiky.

1.14.1 Test stacionárnosti

Pro reprodukovatelnost daných fyzikálních procesů je důležitá jejich stacionárnost. Když je uvažovaný proces náhodný (pravděpodobnostní ve své podstatě), pak je charakterizován pravděpodobnostním rozdělením příslušných proměnných. Důležitou charakteristikou stacionárnosti je pak časová nezávislost příslušných pravděpodobností. To vyžaduje, aby parametry systému zůstávaly konstantní v čase a příslušný jev byl dobře vzorkovatelný.

Problém určení stacionárnosti procesu je obecně velmi komplikovaný. V našem případě se však stacionárnost dá garantovat volbou fyzikálního systému a udržováním podmínek pro daný experimentální set up. Důležitým kritériem je, aby časový experimentální záznam byl mnohem delší než je typická časová škála chování systému. Prakticky se pak postupuje tak, že se měří a určují základní statistické veličiny pro několik segmentů experimentálních dat. Konkrétně se jedná např. o pravděpodobnosti přechodu, korelace, rozptyl, přičemž se porovnávají hodnoty těchto veličin pro první a druhou polovinu dat.

1.14.2 Test na determinismus

Problém determinismus versus náhoda je jedním z kardinálních problémů i současné vědy. Historicky můžeme hovořit o jakési zajímavé hře mezi determinismem a náhodou. Nové světlo do tohoto velmi starého problému mohla vnést až současná nelineární věda, především pak teorie deterministického chaosu. Objevit determinismus v daném procesu a jednoznačně potvrdit jeho význam při vzniku velice komplikovaných řešení v jednoduchých, ale silně nelineárních systémech je doposud v obecné rovině problém otevřený. Prakticky se proto postupuje tak, že buď se předpokládá, že nelinearita je malou perturbací určitého lineárního stochastického procesu, anebo naopak, stochastický element se bere jako malá kontaminace deterministického nelineárního procesu. Jedním z důležitých testů na přítomnost determinismu je pak verifikace určitého stupně predikovatelnosti v chování systému. Pokud je systém ad hoc náhodný, predikce není možná.

Jako další test může posloužit tzv. rekonstrukce atraktora z naměřených dat, která se realizuje pomocí metody vnoření do více dimensionálních fázových prostorů. Procedura je technicky náročná, ale v poslední době byly vypracovány metody její realizace. V tomto případě odpovídá ad hoc náhodnému procesu homogenní atraktor. Jinak se bude jednat o systém s deterministickým chováním [5].

Literatura:

[1] "Kompletace, oživení a statistické testy generátoru náhodné nezávislé binární posloupnosti GENAP V č.0001 - 0003".
Příloha k č.j.: Š - 207/200-78

[2] Peres, Y.: "Iterating von Neumanns Procedure For Extracting Random Bits."
The Annals of Statistics, 1992, Vol..20, No. 1, pp. 590-597.

[3] Knuth, D. E.: "The Art of Computer Programming - Volume 2"
1969, Addison-Wesley Publishing Company

[4] Dawson, E.: "CRYPT - XS, Statistical package for stream ciphers".
Queensland University of Technology, Information Security Research Centre.

[5] Kantz, H. & Schreiber, T.: „Nonlinear Time Series Analysis“.
Cambridge UP, 2000.