VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020 Bc. Matúš Bafrnec

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

HODNOCENÍ ZDROJŮ ENTROPIE V BĚŽNÝCH POČÍTAČÍCH
EVALUATION OF ENTROPY SOURCES IN COMMON COMPUTERS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Matúš Bafrnec

VEDOUCÍ PRÁCE
SUPERVISOR
doc. Ing. Karel Burda, CSc.

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Matúš Bafrnec ***ID:*** 186025
Ročník: 2 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Hodnocení zdrojů entropie v běžných počítačích

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a popište problematiku zdrojů entropie a jejich hodnocení. Na tomto základě zrealizujte programy pro získávání bitových posloupností ze zdrojů entropie, které jsou dostupné v běžných počítačích. Zrealizujte rovněž program pro hodnocení zdrojů entropie, který bude založen na standardu NIST SP 800-90B. Pomocí svých programů zjistěte základní parametry zkoumaných zdrojů entropie a podle získaných hodnot jednotlivé zdroje ohodnoťte.

DOPORUČENÁ LITERATURA:

[1] Turan M. S. aj.: Recommendation for the entropy sources used for random bit generation. NIST SP 800-90B. Gaithersburg: National Institute of Standards and Technology, 2018. Dostupné na: https://bit.ly/2NrBGI8

[2] Burda K.: Kryptografické generátory. Sdělovací technika, 2016, č. 6.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 1.6.2020

Vedoucí práce: doc. Ing. Karel Burda, CSc.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Táto práca sa zaoberá problematikou zdrojov entropie a ich hodnotenia. Obsahuje stručný úvod do teórie informácie, popis zdrojov entropie, ich parametrov a vlastností a spôsobu ich hodnotenia na základe štandardu organizácie NIST SP 800-90B. Ďalšia časť práce je venovaná dvom vytvoreným programom a hodnoteniu a porovnaniu samotných zdrojov entropie. V závere je venovaný priestor použitiu hešovacích funkcií v spojení so zdrojmi entropie.

KĽÚČOVÉ SLOVÁ

entropia, zdroj, hodnotenie, NIST, sieť, zvuk, obrazovka, USB

ABSTRACT

This thesis is focused on entropy sources and their evaluation. It includes a brief introduction to the information theory, description of entropy sources, their parameters and characteristics and methods of evaluation based on the NIST organisation standard SP 800-90B. The following part of the thesis is dedicated to the description of two created programs and evaluation and comparison of entropy sources. Additionally, the last part describes the usage of hash functions in association with entropy sources.

KEYWORDS

entropy, source, evaluation, NIST, network, audio, screen, USB

BAFRNEC, Matúš. *Hodnocení zdrojů entropie v běžných počítačích*. Brno, 2020, 92 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Karel Burda, CSc.

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Hodnocení zdrojů entropie v běžných počítačích“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno . . . .

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce, pánovi doc. Ing. Karlu Burdovi, CSc. za odborné a zároveň priateľské vedenie, konzultácie, trpezlivosť a voľné ruky pri smerovaní práce. Veľká vďaka patrí takisto rodine a priateľom za ich podporu počas celého štúdia.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam výpisov

Úvod

Každý bežný počítač obsahuje zdroj ***neurčitosti*** (odb. zdroj entropie), ktorý systému umožňuje získavať prúd náhodných bitov. Aby bolo možné zaručiť, že poskytované bity sú naozaj náhodné, je nutné tieto zdroje dôkladne poznať, pravidelne testovať a hodnotiť. Len po splnení týchto požiadaviek je možné o bitoch získaných zo zdroja entropie povedať, že sú z pohľadu cieľovej aplikácie nepredvídateľné.

Táto práca pozostáva zo štyroch hlavných častí, z ktorých je každá venovaná inej problematike zdrojov entropie. Prvá kapitola práce sa zameriava na definíciu samotnej entropie v informačnej teórii, jej výpočtu a významu. Následne sú popísané všetky komponenty zdroja entropie, pričom pozornosť je venovaná hlavne zdrojom šumu v počítačoch, popisu princípov ich fungovania a ich parametrom.

V druhej kapitole sa čitateľ dozvie o spôsoboch hodnotenia zdrojov entropie. V úvode sú popísané požiadavky kladené na proces hodnotenia, nasledované popisom testov spoľahlivosti a spôsobov rozhodovania o nezávislosti dát. Poslednú časť kapitoly tvorí popis postupov pre odhad entropie na výstupe zdroja.

Tretia kapitola je venovaná popisu dvoch zrealizovaných programov – **programu pre získavanie bitových postupností zo zdrojov entropie** a **programu pre hodnotenie zdrojov entropie**. V rámci popisu sú uvedené požiadavky na spustenie jednotlivých programov, koncepty, princípy, použité knižnice a technológie a na záver snímky z reálneho použitia programov.

Posledná kapitola sa zaoberá hodnotením zdrojov entropie, z ktorých je možné v rámci vytvoreného programu získať výstupné postupnosti. V úvode kapitoly je špecifikovaná metodika zbierania údajov a popísané parametre zdrojov v rôznych podmienkach. Následne sú vlastnosti zdrojov zovšeobecnené a porovnané na základe vytvorenej veličiny – **výdatnosti**. V závere je venovaný krátky priestor aj hešovacím funkciám v spojení so zdrojmi entropie.

1 Zdroje entropie v počítačoch

Hod kockou v hre, simulácie pomocou metódy Monte Carlo, generovanie kryptografických kľúčov. Všetky tieto procesy spája požiadavka určitého zdroja náhody, ktorý je kľúčový pre jeho samotné fungovanie [1]. V istých prípadoch ide len o férovosť hry, v tých dôležitejších sa jedná o presnosť simulácie alebo až o bezpečnosť údajov a dát. Z týchto (a mnohých ďalších) dôvodov vznikajú požiadavky na kvalitné zdroje neurčitosti – **entropie** – v počítačoch, ktoré dokážu poskytnúť dostatočne rýchly tok nepredvídateľný bitov.

1.1 Entropia

Entropia bola prvýkrát definovaná v roku 1865 ako fyzikálna veličina, ktorá určuje neusporiadanosť, resp. neurčitosť systému [2]. V teórii informácií tento pojem v roku 1948 prevzal *Claude E. Shannon*. Informácia, tak ako ju definoval *Shannon*, nám prináša zníženie neurčitosti systému [3]. Pre množstvo I_i informácie obsiahnuté v správe z_i s pravdepodobnosťou výskytu p_i platí vzťah 1.1 [4]:

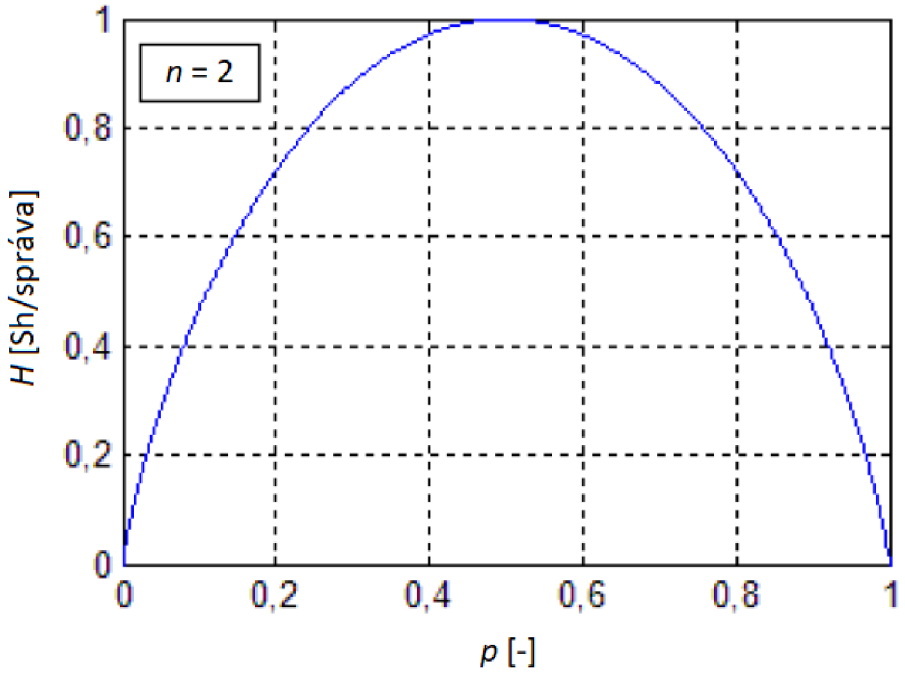
$$I_i = -log_2\ p_i \qquad [\text{Sh} = \text{Shannon}]. \tag{1.1}$$

Z predchádzajúceho vzťahu je možné vidieť, že čím menšia je pravdepodobnosť výskytu správy, tým väčšie množstvo informácie v sebe nesie – často vyskytujúce sa správy v sebe nesú minimálne množstvo informácie a naopak, zriedkavo vyskytujúce sa správy obsahujú väčšie množstvo informácie. Pre zdroj Z správ z_1, z_2, ..., z_n, ktoré sa vyskytujú s pravdepodobnosťou p_1, p_2, ..., p_n je definovaná **entropia zdroja** H vzťahom 1.2 [5] ako stredná hodnota množstva informácie I_1, I_2, ..., I_n jednotlivých správ:

$$H = \sum_{i=1}^{n} p_i \cdot I_i = -\sum_{i=1}^{n} p_i \cdot log_2\ p_i \qquad [\text{Sh/správa}], \tag{1.2}$$

kde n je počet prvkov množiny správ. Pre binárne správy ($n = 2$) je na obrázku 1.1 zobrazená závislosť entropie H na pravdepodobnosti p výskytu zvolenej binárnej správy (tzn. 1 alebo 0) na výstupe zdroja.

Medzi dôležité pojmy patrí aj **minimálna entropia**, ktorá určuje minimálne množstvo informácie (z nášho pohľadu minimálne množstvo ***neurčitosti***) v ľubovoľne zvolenej správe zdroja – množstvo informácie obsiahnuté v každej správe, ktorá sa na výstupe zdroja môže objaviť, je väčšie alebo rovné minimálnej entropii. Je daná minimálnou hodnotou množstva informácie obsiahnutého v správach s pravdepodobnosťou výskytu p_1, p_2, ..., p_n. Matematicky je definovaná vzťahom 1.3, kde



Obr. 1.1: Graf závislosti entropie na pravdepodobnosti výskytu binárnej správy.

funkcie *min* a *max* slúžia k nájdeniu minimálnej, resp. maximálnej hodnoty z danej množiny pre i idúce od 1 po n:

$$H_{\min} = \min_{i=1}^{n}(-log_2\ p_i) = -log_2\ \max_{i=1}^{n}(p_i) \qquad [\text{Sh/správa}]. \tag{1.3}$$

Keďže entropia je mierou neurčitosti systému (z fyzikálnej definície, ktorá sa v tomto prípade aplikuje rovnako aj v informačnej teórii), najvyššia bude v prípade, kedy je výskyt každej správy rovnako pravdepodobný a pre pozorovateľa náhodný. To nás privádza k pojmu **maximálna entropia**, ktorú zdroj dosahuje v situácii, kedy sa všetky jeho správy vyskytujú s rovnakou pravdepodobnosťou $p = \frac{1}{n}$ a sú na sebe štatisticky navzájom nezávislé. Definovaná je vzťahom 1.4 [5]:

$$H_0 = -\sum_{i=1}^{n} p_i \cdot log_2\ p_i = -\sum_{i=1}^{n} \frac{1}{n} \cdot log_2\ \frac{1}{n} = -log_2\ \frac{1}{n} = log_2\ n \qquad [\text{Sh/správa}]. \tag{1.4}$$

Alternatívne sa dá maximálna entropia popísať ako minimálny počet otázok typu áno/nie, ktorý je potrebný k uhádnutiu náhodne zvolenej správy. Cieľom každého zdroja entropie je poskytnúť čo najväčšiu možnú mieru entropie, ktorá pri ideálnom binárnom zdroji dosahuje hodnotu $H = 1$ Sh/správa. V takom prípade hovoríme o skutočne náhodne generovaných bitoch, ktoré nie je žiadnym spôsobom možné predvídať. Je dôležité poznamenať, že všetky vzťahy pre entropiu majú jednotku

Shannon na správu. V prípade zdrojov entropie popisovaných v tejto práci sa pod správou rozumie jeden znak (binárny či nebinárny) na výstupe zdroja. V prípade digitálneho binárneho zdroja je znakom jeden bit. V nasledujúcom texte budú pojmy **správa** a **znak** (pre nebinárny zdroj, resp. **bit** pre binárny zdroj) považované za synonymá.

1.2 Zdroje entropie

V bežných počítačoch existuje niekoľko druhov procesov, ktorých správanie je bez veľmi podrobnej znalosti interného stavu systému náročné popísať a predpovedať, čím môžu slúžiť ako zdroje šumu, priame zdroje entropie alebo ako zdroje semien pre entropické generátory. Množstvo z nich je priamo viazaných na aktivitu užívateľa, ktorá je častokrát pre vonkajšieho pozorovateľa takmer nepredvídateľná a náhodná, čo vo výsledku spôsobuje skvalitnenie výstupnej entropie.

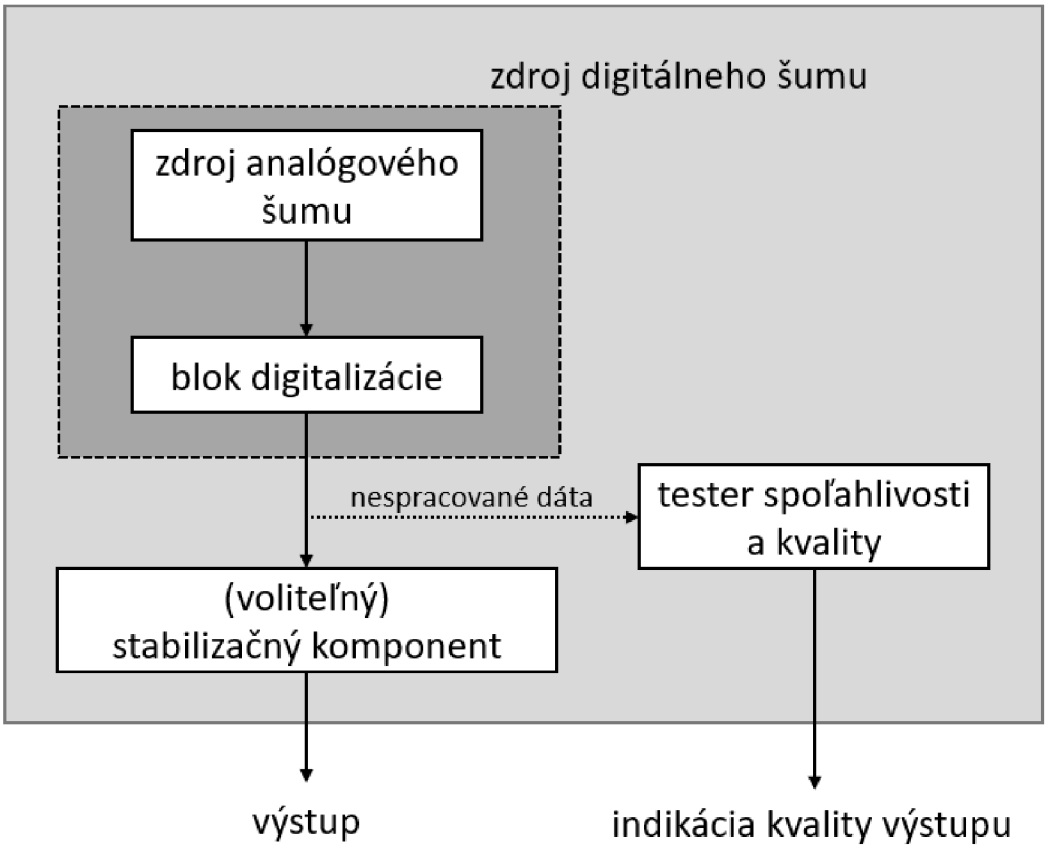
1.2.1 Model zdroja entropie

Model zdroja entropie definovaný v štandarde SP 800-90B [6] organizácie NIST (národný inštitút pre štandardy a technológie – *National Institute of Standards and Technology*) pozostáva z troch základných častí – zdroja šumu, stabilizačného komponentu (voliteľná časť) a testeru spoľahlivosti a kvality. Zjednodušenú schému modelu zdroja entropie je možné vidieť na obrázku 1.2.

Zdroj šumu

Časť bežne dostupných zdrojov entropie v počítačoch je založených na fyzických zdrojoch šumu (samostatný kus hardvéru), ktoré pracujú na princípe prevodu analógového signálu (šumu), ktorý zo svojej podstaty nie je konštantný, na digitálnu hodnotu, ktorá je ďalej spracúvaná. Zvyšné zdroje, založené na ne-fyzických (softvérových) zdrojoch šumu, sa spoliehajú na veľmi presné meranie času medzi (pre vonkajšieho pozorovateľa) náhodnými internými procesmi alebo na ťažko predvídateľný obsah niektorých pamätí a súborov [7].

Pokiaľ je zdrojom šumu proces alebo jav, ktorý priamo negeneruje binárny výstup, musí zdroj entropie obsahovať aj digitalizačnú časť, ktorá sa postará o prevod vstupných vzoriek na bity. Výstupom digitalizovaného zdroja šumu sú takzvané surové dáta (anglicky *raw data*). Pokiaľ zdroj šumu zlyhá alebo prestane generovať náhodný výstup, žiadna iná časť nedokáže kompenzovať neprítomnosť entropie na výstupe.



Obr. 1.2: Model zdroja entropie podľa štandardu NIST SP 800-90B [6].

Stabilizačný komponent

Voliteľný stabilizačný komponent predstavuje deterministickú funkciu, ktorej úlohou je redukcia skreslenia postupnosti a zvyšovanie množstva a kvality entropie na výstupe. Existuje viacero spôsobov, ktorými je možné tieto výsledky dosiahnuť. Podľa odporúčania organizácie NIST je možné použitie kryptografických algoritmov deliacich sa do dvoch skupín – schválených a neschválených. Práve druhá menovaná skupina však nezaručuje, že zdroj bude na výstupe poskytovať udávanú entropiu, preto je nutné jeho funkčnosť overiť definovanými testami [6, 8].

Medzi kryptografické algoritmy schválené podľa štandardu SP 800-90B patria [6]:

- **HMAC** (autentizačný kód správy založený na heši – *Hash-based Message Authentication Code*) s ľubovoľnou schválenou hešovacou funkciou,
- **CMAC** (autentizačný kód správy založený na šifre – *Cipher-based Message Authentication Code*) s blokovou AES (štandard pokročilého šifrovania – *Advanced Encryption Standard*) šifrou,
- **CBC-MAC** (autentizačný kód správy založený na reťazení šifrových blokov –

Cipher Block Chaining Message Authentication Code) s blokovou AES šifrou,
- akákoľvek schválená hešovacia funkcia,
- **Hash_df** definovaná v dokumente SP 800-90A [8] s ľubovoľnou schválenou hešovacou funkciou,
- **Block_Cipher_df** definovaná v dokumente SP 800-90A [8] s blokovou AES šifrou.

Tester spoľahlivosti a kvality

Testy vykonávané testerom spoľahlivosti a kvality predstavujú dôležitú časť, ktorá je schopná zachytiť zmeny správania zdrojov šumu v čo najkratšom čase a s vysokou mierou spoľahlivosti ich indikovať na výstupe, ktorý je počas činnosti zdroja neustále monitorovaný. Príčinou takejto zmeny alebo zlyhania môže byť napríklad výrobná vada, zmeny okolitých podmienok, kolísanie napájacieho zdroja alebo pokus o útok. Nakoľko je zdroj šumu najdôležitejšou časťou, od ktorej sa odvíja funkčnosť celého zdroja entropie a nadväzujúcich entropických generátorov, je sledovanie jeho správnej funkčnosti povinnou súčasťou každého zdroja entropie. Ich bližší popis je možné nájsť v časti 2.2.

1.2.2 Parametre zdrojov entropie

Existuje niekoľko parametrov každého zdroja, ktoré ovplyvňujú jeho spoľahlivosť, bezpečnosť a použiteľnosť. Medzi najdôležitejšie z nich patria:

- Kvalita entropie – rôzne zdroje produkujú rôzne kvalitnú entropiu. Jedným zo znakov kvalitného zdroja je entropia výstupnej bitovej postupnosti, ktorá sa blíži hodnote 1 Shannon na bit.
- Bitová výdatnosť zdroja – množstvo bitov, ktoré dokáže zdroj vygenerovať za jednu sekundu. Generátor by mal byť schopný dodávať dostatočné množstvo bitov, ktoré nebude v systéme spôsobovať čakanie procesov.
- Stabilita – spôsob, akým sa zdroj správa v rôznych prostrediach. Ideálny zdroj by nemal byť vôbec alebo len minimálne ovplyvnený okolnými podmienkami. Táto vlastnosť je dôležitá hlavne z bezpečnostného hľadiska, kedy útočníkom umelo vytvorené podmienky môžu znížiť kvalitu výstupnej entropie.
- Doba inicializácie zdroja – čas, ktorý je potrebný k stabilizácii zdroja a začiatku generovania dostatočne kvalitnej entropie. Cieľom je znížiť túto hodnotu na minimum.
- Odolnosť voči modifikácii a útokom – zdroj entropie by mal byť schopný zachovať si svoju funkčnosť a požadovanú kvalitu aj za umelo vytvorených podmienok či pri pokusoch o útok alebo modifikáciu. V prípade, že je zdroj možné

útočníkom jednoducho upraviť alebo znefunkčniť, a tým znížiť množstvo entropie na jeho výstupe, je považovaný za nevyhovujúci a vytvára potenciálny vstupný vektor útoku.

1.3 Zdroje šumu

Na nasledujúcich stránkach budú popísané a analyzované možné zdroje šumu v bežných počítačoch z hľadiska ich praktickej použiteľnosti, spoľahlivosti a bezpečnosti. Popis bude zameraný na ich použiteľnosť v zdrojoch entropie a budú sledované parametre popísané v predošlej časti textu 1.2.2.

1.3.1 Klávesnica a myš

Klávesnica a myš patria medzi zdroje entropie, ktoré sú v bežných softvéroch používané už dlhú dobu. Sú jednoduché na implementáciu a poskytujú zdroj neurčitosti, ktorý vychádza z aktivity užívateľa. Implementácia spočíva v zaznamenávaní akcií a časov ich vykonania, z ktorých sú následné odvodené bity, ktoré sa dajú považovať za dostatočne náhodné. Výhodou klávesnice a myši je, že neobsahujú komponenty, ktoré by svojim starnutím viditeľne ovplyvňovali funkčnosť zariadenia, čím sa kvalita entropie nemení ani degradáciou súčiastok a zároveň sú tieto komponenty schopné generovať entropiu takmer ihneď po pripojení k počítaču. Ďalej sú odolné voči modifikácii, nakoľko by tento zásah výrazne poznačil ich funkčnosť, avšak existujú spôsoby, ktorými je možné výstupy z týchto zariadení zachytávať [9].

Problematickým bodom klávesnice a myši je ich neprítomnosť u niektorých typoch počítačov, ako napríklad servery, a kolízia viacerých procesov, ktoré ich ako zdroj náhody zdieľajú. Samotným zdrojom neurčitosti je ľudský faktor, ktorý nie je možné predvídať, avšak ktorý zároveň predstavuje aj slabé miesto samotného zdroja, a to v prípadoch, kedy užívateľ pre urýchlenie procesu miesto bežnej aktivity drží jednu klávesu alebo opakuje rovnaký pohyb myšou. Bitová výdatnosť teda závisí od samotnej aktivity užívateľa.

1.3.2 Sieťová karta

Každý bežný počítač v súčasnej dobe poskytuje možnosť pripojenia k internetu. Operačné systémy sú komplexné systémy, ktoré okrem činností vykonávaných užívateľom generujú množstvo aktivity na pozadí, čo sa odráža na sieťovej prevádzke, ktorá môže slúžiť ako zdroj neurčitosti. Zo všetkých dostupných údajov je možné ako náhodné bity použiť údaje z času príchodu paketov, položky z hlavičky paketu (napr. zdrojová IP adresa, veľkosť paketu, použitý protokol, kontrolný súčet

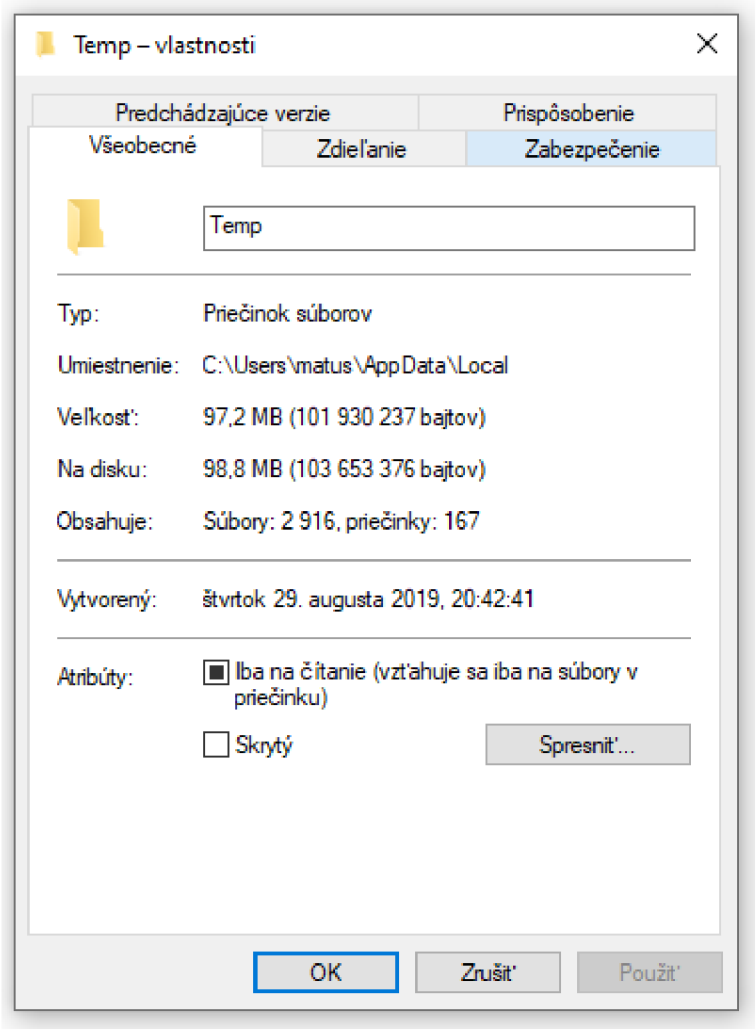
a iné), samotný dátový obsah paketu, poprípade kombináciu všetkých uvedených spôsobov. Práve čas príchodu paketov sa dá považovať za nepredvídateľný a odolný voči útoku, nakoľko závisí od značného množstva sieťových komponentov po trase paketu, nastavenia ich priorít a samotného vyťaženia siete a systému.

V prípade dostatočne bohatej sieťovej prevádzky predstavuje sieťová karta dostatočne kvalitný zdroj šumu, ktorého bitová výdatnosť je aj bez priamej aktivity užívateľa dostatočná. Sieťová aktivita časom nestráca svoju rôznorodosť, preto sa dá považovať za stabilný zdroj entropie. Zo svojej podstaty by modifikácia sieťovej karty bola jednoducho odhaliteľná, problémom však môže byť lokálny útočník, ktorý je schopný sieťovú prevádzku odchytávať a jej obsah analyzovať, poprípade ju priamo generovať. Na druhej strane poskytuje sieťová prevádzka výhodu, ktorou je nepravidelnosť samotnej komunikácie, čo čiastočne eliminuje možnosť zníženia kvality entropie vplyvom ľudského faktoru.

1.3.3 Obsah dočasných súborov na disku

Každý operačný systém si pre vlastné potreby alebo potreby niektorých aplikácií či programov udržiava priečinok s dočasnými súbormi, ktoré by mohli poslúžiť ako zdroj náhody. V systémoch Windows sa jedná o priečinok `Temp`, v systémoch Linux o priečinok `/tmp` alebo `/var/tmp`. Priečinok `/tmp` je často vyprázdňovaný samotným systémom (aj vždy po jeho štarte) a je určený pre súbory s krátkou životnosťou, čo z neho tvorí bezpečnejší, ale menej spoľahlivý zdroj náhody. Naproti tomu `/var/tmp` si zachováva svoj obsah aj po reštartovaní systému a je určený pre trvácnejšie súbory, u ktorých je frekvencia zmien nižšia.

Prvým problémom týchto súborov ako zdrojov entropie je fakt, že sa jedná o čítanie zo súborov vytvorených inými aplikáciami či systémom, čo znamená, že je tretia strana schopná ovplyvniť ich obsah a tým aj výslednú kvalitu entropie. Obsah samotných priečinkov a frekvencia zmeny súborov v nich je silne závislá od aktivity užívateľa a systému, pričom môže trvať relatívne dlhú dobu aj prvotné naplnenie priečinka dátami. Ďalšími faktormi vplývajúcimi na spoľahlivosť obsahu dočasných súborov ako zdroja entropie je jednoduchá možnosť zásahu alebo vytvorenia dočasných súborov útočníkom a zdieľanie priečinka naprieč celým systémom. Navyše, priečinok nie je žiadnym spôsobom chránený proti čítaniu inými aplikáciami. Ako zdroj náhodných bitov môžu slúžiť vybrané bity z veľkosti priečinka a obsah alebo podrobnosti samotných súborov. Na obrázku 1.3 je možné vidieť podrobnosti užívateľského priečinka `Temp` v operačnom systéme Windows 10.



Obr. 1.3: Podrobnosti užívateľského priečinka `Temp` v operačnom systéme Windows 10.

1.3.4 Systémové parametre a premenné

Stav každého systému je popísaný množstvom premenných veličín, ako napríklad aktuálny čas, množstvo voľnej pamäte RAM (pamäť s náhodným prístupom – *Random Access Memory*), obsah pamäte RAM, aktuálne zaťaženie procesora, časovanie (plánovanie) procesov a mnoho iných. Bežné operačné systémy generujú na pozadí množstvo aktivity, ktorá priamo ovplyvňuje spomínané systémové parametre, čím môžu poslúžiť ako zdroj náhody. Iné z nich, ako napríklad fyzická adresa sieťovej karty, výrobné číslo základnej dosky alebo sériové číslo procesoru môžu slúžiť ako unikátne prvky v stabilizačných komponentoch.

Všetky sú dostupné hneď po štarte systému a pravidelne sa aj bez aktivity užívateľa menia. Potenciálnym problémom môže byť ich dostupnosť aj pre ostatných

užívateľov systému, čo je však vyvážené veľkým množstvom parametrov, z ktorých je možné zdanlivo náhodné bity čerpať. Výslednú kvalitu entropie tohto zdroja je možné výrazne zvýšiť aktivitou systému a užívateľa.

1.3.5 Tepelný šum

Meranie teploty počítačových komponentov predstavuje proces, ktorý je na súčasnom hardvéri zaťažený teplotným šumom. Práve tento teplotný šum nám dokáže poslúžiť ako zdroj šumu. Teplota patrí medzi bežne sledované parametre procesoru, grafickej karty, disku a iných komponentov. Jedná sa o hodnotu, ktorá je dostupná už od štartu systému, mení sa s aktivitou užívateľa a je odolná voči útokom. Moderný hardvér a systémy dokážu pomocou regulácie otáčok ventilátorov chladiča udržiavať relatívne stabilnú teplotu každého spomenutého komponentu. Zmeny teploty sú vyvolané zmenami v aktivite systému a jej náročnosti a neistotou samotného merania teploty či teplotného šumu na rezistoroch. Medzi nevýhody teda patrí malá rýchlosť zmeny teploty, čo umožňuje využitie iba niekoľkých posledných bitov z každého merania. Kvalita entropie môže byť časom ovplyvnená degradáciou teplotných čidiel.

1.3.6 Otáčky magnetického disku a ventilátoru

K regulácii teploty počítačových komponentov sú využívané chladiče, ktoré sú vo veľkej časti prípadov aktívne (tj. vybavené ventilátorom). Jeho otáčky sú regulované k udržaniu konštantnej teploty, čo vytvára ďalší zdroj informácie, ktorú je veľmi náročné predpovedať. Teplota a s ňou spojené otáčky chladiacich ventilátorov sa v prípade nízkej aktivity systému udržujú na takmer stabilnej hodnote, čo pre zdroj entropie znamená použitie iba malého počtu posledných bitov zo sledovaných hodnôt. V prípade prítomnosti magnetického disku je možné sledovať drobné výchylky v zmenách rýchlosti otáčania diskov voči výrobcom stanovenej hodnote, čo predstavuje ďalší zdroj zdanlivo náhodných bitov, rovnako ako v prípade otáčok ventilátoru.

1.3.7 Obrazový výstup a webkamera

Obrazový výstup – alebo inak povedané zobrazovaný obraz na monitore počítača – predstavuje bitovo veľmi výdatný zdroj náhody. Aktivita užívateľa tento obraz neustále mení, čo vytvára priestor na získavanie veľkého množstva nepredvídateľných bitov. Obraz je dostupný takmer okamžite po štarte systému a jeho prípadná modifikácia útočníkom je ľahko detegovateľná, pričom výslednú entropiu ovplyvní len minimálne.

Kvalita entropie takéhoto zdroja závisí od konkrétneho získavania bitov z obrazu. Prvá možnosť zahŕňa vytváranie priebežných snímok alebo krátkych záznamov obrazu, čo zabezpečí dostatočnú rôznorodosť získaných bitov pre neskoršie použitie. Problémom je dlhšia neaktivita užívateľa, kedy by uložené bity s dostatočnou entropiou mohli by nahradené bitmi zo statického obrazu.

Druhá možnosť spočíva taktiež vo vytváraní snímok alebo záznamu obrazu, ktorý by v tomto prípade bol zachytený na požiadanie. Samotný vznik tejto požiadavky predstavuje aktivitu v systéme, ktorá by behom krátkej chvíle bola schopná vygenerovať dostatočné množstvo potrebných bitov.

Tretia možnosť využíva odlišný prístup k spracovaniu obrazu, kedy by miesto získavania bitov z jeho záznamu bol počítaný rozdiel voči referenčnej snímke, ktorá by bola náhodne a priebežne menená. Tento prístup by dokázal vygenerovať nepredvídateľné bity aj v prípade menej intenzívnej aktivity užívateľa a zároveň by samotná obrazová informácia nebola dostatočná pre odhalenie generovaných bitov útočníkom.

Zdrojom obrazu nemusí byť iba samotný obrazový výstup, ale aj pripojená webkamera. Tá má oproti detailnému obrazu z monitoru výhodu v obrazovom šume, ktorý zvyšuje entropiu bitového toku zo snímaného obrazu. Jedným z možných útokov na kameru je jej prekrytie, kedy by prišlo k takmer úplnému znehodnoteniu snímaného obrazu a ustáleniu obsahu dátového toku.

1.3.8 Zvuková karta

Väčšina moderných počítačov je vybavená integrovaným alebo externým mikrofónom, ktorý prevádza okolitý zvuk na analógový signál, ktorý je ďalej vzorkovaný, kvantovaný a prevedený na digitálnu hodnotu. Každý prevod zvuku je zaťažený šumom z okolia, ktorý za bežných podmienok nie je možné kompletne eliminovať a spolu s hlukom a zvukmi okolia vytvárajú neperiodický a nepredvídateľný zdroj informácií. Tento fakt, spolu s neistotou prevodu zvuku na digitálny signál, predstavujú zdroj entropie, ktorý je dostupný hneď po štarte systému, bitovo výdatný, stabilný a odolný voči modifikácii.

Spôsob zaznamenávania bitov spracovaných zvukovou kartou je takisto možné riešiť tromi spôsobmi, ako v prípade obrazového výstupu v časti 1.3.7. Porovnávanie záznamu s referenčnou nahrávkou by pridalo ďalší stupeň bezpečnosti voči útoku a zároveň by zvýšilo kvalitu získanej entropie. Tú takisto ovplyvňuje aj okolité prostredie, ktoré v prípade okolitého ruchu privádza na vstup rôznorodejší signál, ktorý sa vo výsledku pozitívne prejaví na množstve entropie vo výslednom bitovom toku.

1.4 Využitie entropie v generátoroch

Zariadenia využívajúce zdroje entropie ku generovaniu toku nepredvídateľných bitov sa nazývajú entropické generátory. Existujú dva druhy generátorov, ktoré sa líšia svojim účelom, fungovaním a štatistickými vlastnosťami. Najčastejšie sú využívané ako zdroje kryptografických kľúčov, kedy je ich správna implementácia a odolnosť voči útoku jednou z kľúčových vlastností.

1.4.1 Generátor náhodných bitov

RBG (generátor náhodných bitov – *Random Bit Generator*), známy aj pod skratkami TRNG (generátor skutočne náhodných čísel – *True Random Number Generator*) alebo HRNG (hardvérový generátor náhodných čísel – *Hardware Random Number Generator*), patrí do triedy entropických generátorov založených na javoch, ktoré nie je možné so súčasnými poznatkami predpovedať. Jedná sa teda o zdroj skutočnej náhody pochádzajúcej z kvantových (napr. rádioaktívny rozpad častíc detegovaný Geiger-Müllerovým detektorom, fotóny prechádzajúce cez polopriepustné zrkadlo, spontánna parametrická konverzia v optických oscilátoroch [10], fluktuácie v energii vákua merané pomocou homodynnej detekcie [11, 12] a iné) alebo z klasických javov (tepelný šum na rezistore, šum z lavínovej diódy, atmosférický šum).

Problémom RBG je ich schopnosť zlyhať alebo degradovať „potichu“, čo sa prejaví na znížení kvalite výstupnej entropie. Príkladom môže byť postupné spomaľovanie rádioaktívneho rozpadu, ktoré je z výstupu generátoru náročné detegovať. Riešením sú testy spoľahlivosti definované štandardom NIST a popísané v časti 2.2, ktoré sú schopné odhaliť zmeny v správaní zdroja šumu.

Medzi zaujímavé projekty poskytujúce RBG patrí napríklad **LavaRnd** [13], ktorý spĺňa požiadavky organizácie NIST a pracuje na princípe zmien v obraze, ktorý je generovaný pohybom v lávových lampách. Projekt **HotBits** [14] rovnako predstavuje generátor skutočne náhodných čísel, ktorý je založený na rádioaktívnom rozpade častíc. Asi najznámejším zo všetkých je generátor **RANDOM.ORG** [15], ktorý ako zdroj entropie využíva atmosférický šum a medzi ktorého výhody patrí webové rozhranie poskytujúce generátor náhodných čísel z daného rozsahu.

1.4.2 Deterministický generátor náhodných bitov

DRBG (deterministický generátor náhodných bitov – *Deterministic Random Bit Generator*), v literatúre často uvádzaný aj ako PRNG (generátor pseudonáhodný čísel – *Pseudo-Random Number Generator*) alebo **generátor pseudonáhodných bitov**, je v skutočnosti algoritmus generujúci postupnosť čísel, ktorá sa svojimi štatistickými vlastnosťami približuje postupnosti skutočne náhodných čísel. Táto

postupnosť nie je skutočne náhodná, nakoľko je závislá od inicializačnej hodnoty, tzv. semena generátoru. Kvalitný generátor pseudonáhodných čísel sa v porovnaní s generátormi skutočne náhodných čísel vyznačuje svojou rýchlosťou, nakoľko nie je blokovaný čakaním na bity zo zdroja entropie, ale je schopný generovať dostatočné množstvo čísel. Samotné množstvo použiteľných čísel je obmedzené kvalitou generátoru, na ktorej sa podieľa aj periodicita – počet čísel, ktoré je generátor schopný vygenerovať bez opakovania výstupu a obnovenia semena [16].

Problémom môžu byť generátory pseudonáhodných čísel s uzavretým zdrojom (angl. *closed source*, opak softvéru s otvoreným zdrojom – angl. *open source*), ktoré sa v minulosti vyznačovali chybami ako napríklad kratšími periódami pre niektoré semená, nerovnomerným rozložením výstupu či koreláciou medzi generovanými číslami, ktoré boli odhalené až po rokoch používania [17]. Príkladom môže byť jazyk Java vo verzii 8, ktorý pre generovanie pseudonáhodných čísel v triede `Random` využíva lineárny kongruentný generátor – *Linear Congruential Generator* (skrátene LCG), ktorý nie je považovaný za kryptograficky bezpečný [18, 19]. V súčasnej dobe dokumentácia obsahuje poznámku o nedostatočnej kryptografickej bezpečnosti a pre kryptograficky citlivé aplikácie odporúča užívateľom zvážiť použitie inej, kryptograficky bezpečnej triedy pre generovanie pseudonáhodných čísel.

2 Hodnotenie zdrojov entropie

Každý zdroj entropie potrebuje vlastnú sadu nástrojov a definovaných testov, ktoré umožňujú kontrolovať jeho správnu činnosť a odhadnúť množstvo entropie na jeho výstupe. Táto kapitola sa bude zaoberať validáciou zdrojov entropie, testami spoľahlivosti, overovaním nezávislej a identickej distribúcie výstupu zdrojov a odhadom minimálnej entropie na výstupe. Poznatky z tejto kapitoly sú prevzaté z odporúčania organizácie NIST SP 800-90B [6].

2.1 Validácia zdroja entropie

Proces validácie je kľúčový pre overenie všetkých požiadaviek kladených na zdroj entropie špecifikáciou SP 800-90B [6]. Validácia spočíva v testovaní nezávislým akreditovaným laboratóriom NVLAP (národný program dobrovoľnej akreditácie laboratórií – *National Voluntary Laboratory Accreditation Program*) voči požiadavkám daným dokumentom SP 800-90B [6]. Úspešné absolvovanie procesu validácie poskytuje uistenie, že zdroj poskytuje adekvátne množstvo entropie, ktoré môže byť vyžadované pre splnenie niektorých právnych či iných požiadaviek.

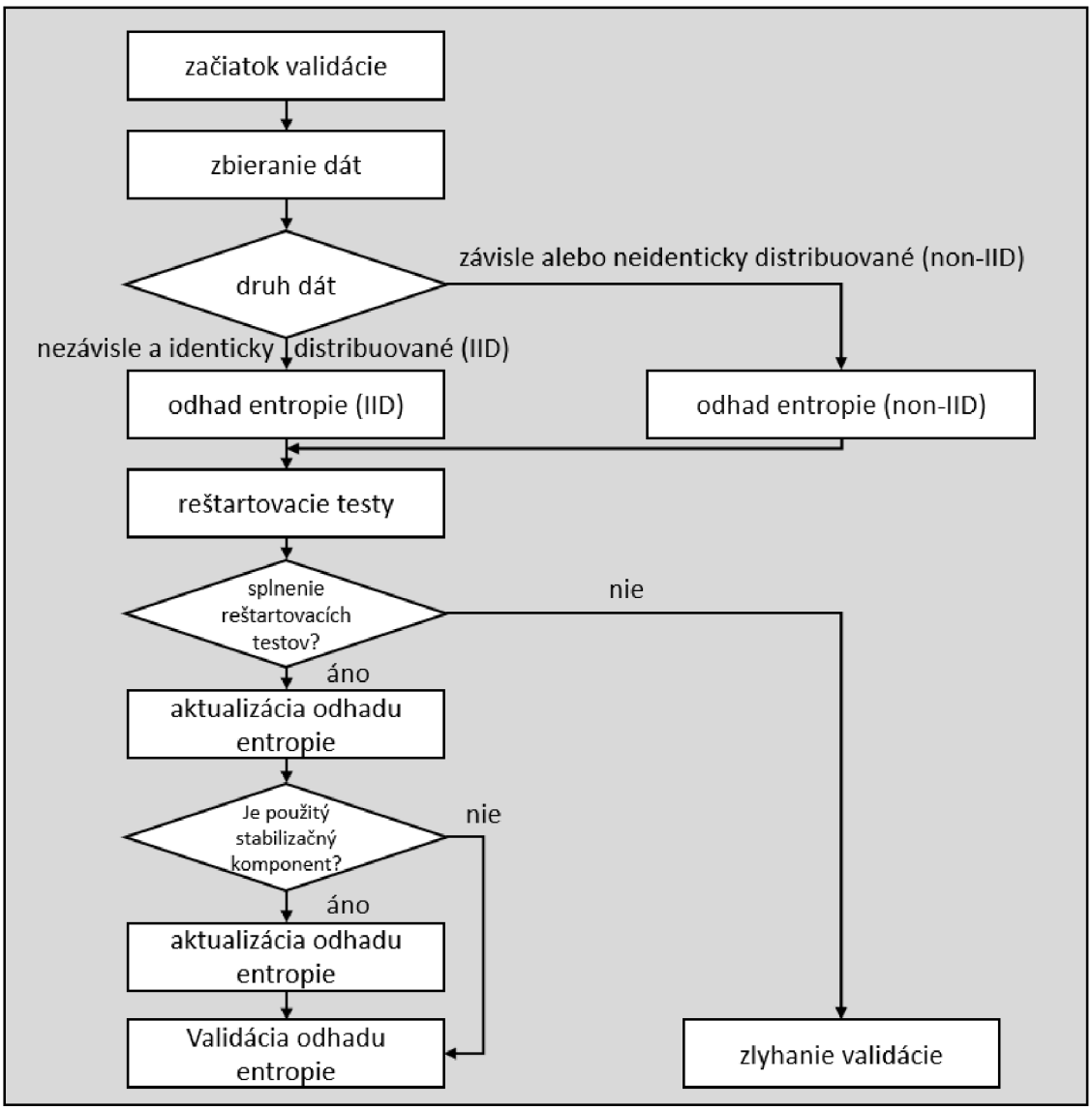
Na začiatku celého procesu je predloženie zdroja entropie akreditovanému laboratóriu. Následne laboratórium preskúma priloženú **dokumentáciu** zdroja, ktorá bude slúžiť ako oporný bod pre celé testovanie. Dokumentácia by mala obsahovať kompletný popis zdroja entropie, ako napríklad popis ideálnych prevádzkových podmienok a ich hraničných stavov, testov pre overenie jeho funkčnosti, ďalšie testy pridané vývojármi, možné prejavy zlyhania či iné vlastnosti zdroja. Laboratórium preverí tvrdenia uvedené v dokumentácii a v prípade potreby požiada o predloženie dôkazov k tvrdeniam či o vysvetlenie prípadných nejasností.

Následne sa laboratórium riadi postupom, ktorý je možné vidieť na obrázku 2.1.

2.1.1 Zbieranie dát

Žiadateľ o validáciu zdroja entropie na začiatok predloží dáta spĺňajúce nasledujúce podmienky:

1. Žiadateľ odovzdá súbor o veľkosti 1 000 000 vzoriek sekvenčne zachytených priamo z výstupu zdroja šumu (tzv. surové dáta). Pokiaľ je získanie 1 000 000 po sebe nasledujúcich vzoriek nemožné, je povolené zlúčenie menších sekvenčne zachytených súborov vzoriek, ktoré by mali obsahovať aspoň 1 000 vzoriek na súbor a spolu by mali dosahovať požadovanej veľkosti súboru.



Obr. 2.1: Proces validácie zdroja entropie.

2. Pokiaľ zdroj entropie obsahuje stabilizačný komponent, súčasťou predložených dát by mal byť súbor vzoriek zo stabilizačného komponentu, spĺňajúci rovnaké podmienky ako súbor vzoriek zo zdroja šumu. Zároveň by k týmto vzorkám malo byť pristupované ako k binárnym dátam. Špecifikácia povoľuje, aby vzorky získané zo zdroja šumu v predošlom bode boli použité ako vstup pre stabilizačný komponent.
3. Dáta pre reštartovacie testy (popísané v časti 2.1.3) by mali byť obstarané po uskutočnení 1 000 reštartov, po každom z ktorých bolo zozbieraných 1 000 vzoriek. Vzorky by mali byť získané v momente, keď je zdroj šumu pripravený poskytovať výstup, ktorý sa premietne do výstupu zdroja entropie. Tieto dáta

sú uložené v reštartovacej matici $\boldsymbol{M}$ s rozmermi 1 000 x 1 000.

2.1.2 Rozhodovanie o druhu dát

Odhadovanie entropie prebieha rozlične pre rôzne druhy dát – IID (nezávisle a identicky distribuované – *Independent and Identically Distributed*) dáta a dáta, ktoré túto podmienku nespĺňajú (*non*-IID). Dáta sa dajú považovať za nezávislé a identicky distribuované iba v prípade splnenia **všetkých** nasledujúcich podmienok:

- žiadateľ o validáciu zdroja na základe analýzy zdroja predloží jeho vyjadrenie o rozložení dát spolu s dôkazmi podporujúcimi jeho tvrdenie,
- súbor vzoriek zo zdroja šumu prejde testami definovanými a popísanými v časti 2.3,
- riadky a stĺpce reštartovacej matice $\boldsymbol{M}$ prejdú testami definovanými a popísanými v časti 2.3,
- ak je použitý stabilizačný komponent, súbor vzoriek z jeho výstupu splní podmienky IID dát kladené testami v časti 2.3.

Pokiaľ nie je niektorá z týchto podmienok splnená, sú dáta z výstupu zdroja entropie považované za závislé alebo neidenticky distribuované (*non*-IID).

2.1.3 Reštartovacie testy

Odhad entropie z jednej dlhej neprerušenej sekvencie môže nadhodnotiť množstvo entropie na výstupe v prípade, že zdroj po reštartovaní generuje korelovaný výstup (jednotlivé znaky na výstupe nie sú vzájomne nezávislé). V prípade prístupu útočníka k viacerým výstupným sekvenciám po reštarte by bol schopný odhadnúť ďalší výstup zdroja s úspešnosťou vyššou, aká prináleží odhadovanej entropii. Dáta pre reštartovacie testy predstavujú jednotlivé riadky a stĺpce reštartovacej matice $\boldsymbol{M}$ dĺžky 1 000 vzoriek, čím sa pri dvojrozmernej matici dostávame k 2 000 súborom vzoriek.

Na začiatku je vykonaný úvodný test, ktorý skúma výskyt najčastejšej hodnoty v riadkoch a v stĺpcoch matice $\boldsymbol{M}$. V prípade, že sa daná hodnota vyskytuje významne častejšie ako by podľa odhadu výstupnej entropie mala, reštartovacie testy zlyhajú a zdroju entropie nie je udelený výsledný odhad výstupnej entropie. V prípade, že úvodný test nezachytí žiadnu významnejšiu odchýlku výskytu najčastejšej hodnoty od odhadu, sú na riadkoch a stĺpcoch matice $\boldsymbol{M}$ vykonané odhady entropie popísané v časti 2.4. Pokiaľ je estimácia entropie podľa riadkov alebo stĺpcov menšia ako polovica entropie v predošlom odhade, tento krok zlyhá a zdroju nie je udelený výsledný odhad výstupnej entropie. Výslednou entropiou vstupujúcou do ďalšieho kroku je minimálny odhad entropie pochádzajúci z predošlého kroku, z odhadu entropie podľa riadkov a z odhadu podľa stĺpcov. Obdobným spôsobom je získavaný

aj výstupný odhad entropie v každom ďalšom kroku – ako minimálna odhadnutá hodnota entropie z predošlého a súčasného kroku.

2.1.4 Stabilizačný komponent

Pokiaľ zdroj entropie obsahuje aj niektorý zo stabilizačných komponentov popísaných v časti 1.2.1, je ešte pred udelením finálneho odhadu výstupnej entropie vykonaný ďalší odhad popísaný v dokumente SP 800-90B [6]. Pri udeľovaní odhadu je potrebné preskúmať, či použitý stabilizačný komponent patrí medzi schválené alebo neschválené. Výsledná entropia celého zdroja je následne menšia z hodnôt získaných v tomto a v predošlom kroku.

2.2 Testy spoľahlivosti zdrojov šumu

Testy spoľahlivosti patria medzi povinné súčasti zdroja entropie. Ich úlohou je detegovať zmeny správania alebo priamo zlyhanie zdroja šumu. Tieto testy sú veľmi úzko zviazané s technológiou konkrétneho zdroja šumu, nakoľko vo väčšine prípadov zdroj neprodukuje neskreslené dáta. Z toho dôvodu sú tradičné štatistické testy nepoužiteľné a je úlohou vývojára navrhnúť testy, ktoré zodpovedajú očakávanému štatistickému správaniu korektne fungujúceho zdroja. Testy by v ideálnom prípade mali byť schopné upozorniť na tieto tri prípady:

1. výrazné zníženie entropie na výstupe zdroja šumu,
2. zlyhanie zdroja šumu,
3. zlyhanie hardvéru a následné nekorektné fungovanie zdroja a jeho súčastí.

2.2.1 Druhy testov

Testy spoľahlivosti by mali byť použité priamo na zdroj šumu, ešte pred stabilizačným komponentom. Špecifikácia SP 800-90B [6] umožňuje použitie testov spoľahlivosti aj na stabilizovaný výstup, avšak táto možnosť nepatrí medzi požiadavky na zdroje entropie. Poznáme tri druhy testov, ktoré budú popísané v nasledujúcich častiach.

Testy by mali byť navrhnuté tak, aby v prípade zlyhania zdroj entropie tento stav oznámil nadradenej aplikácii. Môže byť definovaných viacero druhov zlyhania, na ktoré môže aplikácia reagovať rôznymi spôsobmi (napríklad dočasné zlyhania nemusia viesť k úplnému zastaveniu využívania zdroja, ale len k jeho pozastaveniu). Takisto je možné definovať kritické hodnoty, ktoré sú príznakom trvalého zlyhania zdroja entropie a v prípade výskytu takejto situácie ukončiť jeho činnosť. Všetky

tieto hodnoty, spolu s úplným popisom zdroja a chybových stavov by mali byť zahrnuté v sprievodnej dokumentácii. Pri návrhu testov by mal byť obzvlášť dôraz na testovanie a detekciu nekorektného správania zdroja na hranici normálnych prevádzkových podmienok a podmienok mimo navrhnutého prevádzkového rozsahu.

Testy po štarte

Testy po štarte sú určené na testovanie zdroja šumu po jeho zapnutí alebo reštarte, ešte pred použitím zdroja entropie. Jeho účelom je overiť korektné fungovanie zdroja šumu za bežných prevádzkových podmienok a zistiť prípadné zlyhanie od posledného testu po štarte. Výstup zdroja šumu **nesmie** byť pred úspešným dokončením testov použitý, po overení očakávaného správania môžu byť zhromaždené bity zahodené alebo použité v ďalších častiach zdroja entropie.

Podľa odporúčania by mali byť testy po štarte zahŕňať priebežné testy vykonané na spojitej vzorke o veľkosti aspoň 1024 znakov (bitov). Okrem priebežných testov je možné doplniť testy po štarte aj o vývojármi definované testy prispôsobené priamo povahe zdroja šumu.

Priebežné testy

Priebežné testy bežia kontinuálne nad výstupom zdroja šumu a ich účelom je detegovať zlyhania počas činnosti zdroja entropie. Priebežné testy môžu zahŕňať niekoľko testov a v závislosti od zdroja spracúvať množstvo bitov za sekundu, preto je na testy kladená požiadavka nízkej pravdepodobnosti vyvolania falošného poplachu počas normálneho fungovania zdroja. Priebežné testy väčšinou pracujú s obmedzenými zdrojmi, čo výrazne znižuje ich schopnosť odhaliť nevýrazné zmeny v správaní zdroja, preto je väčšina z nich schopná odhaliť len **závažné zlyhania**.

Od testov po štarte sa líšia tým, že pracujú nad prúdom bitov a neblokujú výstup počas trvania testu. Preto je potrebné brať do úvahy, že zdroj môže istý čas produkovať výstup s nedostatočným množstvom entropie pred samotným odhalením zlyhania zdroja šumu, ku ktorému je vzhľadom na nastavenia sily testu potrebné zozbierať dostatočné množstvo dôkazov o zlyhaní. Aby sa predišlo falošným detekciám zlyhania zdroja, je potrebné zvýšiť silu testu natoľko, aby sa pravdepodobnosť falošného poplachu znížila pod požadovanú úroveň. Pri nastavovaní sily testu je takisto potrebné brať do úvahy aj bitovú rýchlosť zdroja (tj. silný test požadujúci množstvo dôkazov môže pri pomalom zdroji výrazne oddialiť detekciu zlyhania).

Testy na vyžiadanie

Testy na vyžiadanie, ako vypovedá samotný názov, môžu byť spustené v akomkoľvek momente. Dokument SP 800-90B [6] nevyžaduje vykonávanie testov na vyžiadanie

počas prevádzky, ale vyžaduje, aby bol zdroj entropie schopný vykonať tieto testy na zdroji šumu. Tieto testy by mali zahŕňať minimálne aspoň rovnaké testy, aké sú použité v testoch po štarte. Medzi prijateľné metódy spúšťania testov na vyžiadanie patrí resetovanie, reštartovanie alebo zapnutie zdroja entropie, avšak iba v prípade, pokiaľ je hneď v zápätí vykonaný test po štarte. Čo sa týka testovaných bitov, výstup zo zdroja šumu **nesmie** byť dostupný až do úspešného ukončenia testov. Nahromadené testované bity môžu byť následne kedykoľvek zahodené alebo po ukončení testov použité.

2.2.2 Schválené testy spoľahlivosti

Odporúčanie SP 800-90B [6] obsahuje dva schválené priebežné testy – **test počtu opakovaní** a **adaptívny proporčný test**. V prípade, že sú oba zahrnuté v súbore testov, nie sú vyžadované žiadne iné testy, odporúča sa však zahrnúť aj testy prispôsobené pre konkrétny zdroj šumu. Test počtu opakovaní aj adaptívny proporčný test sú navrhnuté tak, aby vyžadovali minimum prostriedkov a bolo možné ich vykonávať za behu zdroja, bez nutnosti čakania výstupu na ich výsledky. Kľúčovým je správne nastavenie parametrov tak, aby pravdepodobnosť chyby prvého typu bola na prijateľnej hranici. V tomto prípade budú testy koncipované tak, aby pravdepodobnosť chyby prvého typu bola $\alpha = 2^{-20}$ (chyba prvého typu znamená zlyhanie testu v prípade korektného fungovania zdroja).

Test počtu opakovaní

Cieľom tohto testu je odhaliť kritické zlyhanie zdroja, ktoré má za následok opakovanie jednej konkrétnej správy (bitu) na výstupe. Zlyhanie testu nastane pri prekročení kritickej hodnoty počtu povolených opakovaní C, ktorá sa odvíja od stanovenej pravdepodobnosti chyby prvého typu. Hodnotu C je možné za pomoci odhadu entropie zdroja H vypočítať prostredníctvom vzťahu 2.1 ako:

$$C = 1 + \left\lceil \frac{-log_2\ \alpha}{H} \right\rceil \qquad [-], \tag{2.1}$$

kde $\lceil ... \rceil$ vráti celé číslo zaokrúhlené nahor. Hodnota C je najmenšie celé číslo spĺňajúce podmienku $\alpha \geq 2^{-H(C-1)}$, ktorá zaručuje, že pravdepodobnosť postupnosti rovnakých hodnôt za sebou v prípade normálneho fungovania zdroja je menšia ako α.

Adaptívny proporčný test

Adaptívny proporčný test je navrhnutý na odhalenie veľkého poklesu entropie zdroja šumu, ktorý môže byť následkom fyzického zlyhania, abnormálnych prevádzkových podmienok alebo pokusu o útok či modifikáciu zdroja. Princípom testu je priebežné

sledovanie výskytu vybranej hodnoty vo výstupnom toku. V prípade jej nadmerného výskytu test zlyhá, čo môže byť indikáciou zlyhania zdroja, ktoré je menej očividné ako to, ktoré dokáže zachytiť test počtu opakovaní.

Test vždy zoberie jednu správu z výstupu a následne sleduje jej výskyt v nasledujúcich $W - 1$ vzorkách, kde W je šírka sledovaného okna. V prípade, že počet výskytov zvolenej správy presiahne kritickú hodnotu C, test vráti chybu a zlyhá. Hodnota C je stanovená tak, aby pravdepodobnosť výskytu zvolenej správy v danej sekvencii bola menšia ako pravdepodobnosť chyby prvého typu α. V prípade binárneho zdroja je možné test upraviť a sledovať aj málo častý výskyt zvolenej správy, ktorý by znamenal nadmerný výskyt druhej správy.

V tabuľke 2.1 a 2.2 je možné vidieť odporúčané [6] kritické hodnoty pre binárny a nebinárny zdroj s veľkosťou okna $W = 1024$, resp. $W = 512$ správ.

Tab. 2.1: Kritická hodnota C v závislosti od entropie zdroja H pre binárny zdroj s veľkosťou okna $W = 1024$ správ.

Entropia zdroja H [Sh/znak]	Kritická hodnota C [–]
0,2	941
0,4	840
0,6	748
0,8	664
1,0	589

Tab. 2.2: Kritická hodnota C v závislosti od entropie zdroja H pre nebinárny zdroj s veľkosťou okna $W = 512$ správ.

Entropia zdroja H [Sh/znak]	Kritická hodnota C [–]
0,5	410
1	311
2	177
4	62
8	13

2.3 Overovanie IID predpokladu

Vzorky zo zdroja šumu je možné považovať za nezávislé a identicky distribuované (IID) v prípade, že má každá vzorka rovnaké rozdelenie pravdepodobnosti ako všetky ostatné vzorky a všetky vzorky sú navzájom nezávislé. V prípade, že vieme dáta prehlásiť za IID, je proces odhadu entropie výrazne jednoduchší. V prípade, že vzorky túto podmienku nespĺňajú, je nutné použiť iné, náročnejšie metódy. Štatistické testy popísané na nasledujúcich stránkach sú navrhnuté tak, aby overili nezávislosť a distribúciu dát a sú prevzaté z odporúčania SP 800-90B [6]. V prípade, že žiadny z testov nezlyhá, sú dáta prehlásené za IID, v opačnom prípade sú považované za *non*-IID.

2.3.1 Permutačné testy

Permutačné testovanie je spôsob testovania štatistických hypotéz, kedy je štatistický parameter T testu porovnávaný so štatistickou distribúciou získanou zo vstupných dát miesto štandardnej štatistickej distribúcie. Permutačné testy pracujú v režime, kedy je vypočítaný štatistický parameter T pre pôvodný súbor dát a počítadlá C_0 a C_1 sú nastavené na hodnotu 0. Následne je vytvorených 10 000 permutácií pôvodného súboru pomocou Fisher-Yatesovho miešacieho algoritmu [20], z ktorých sú vypočítané nové štatistické parametre T_i (kde i je poradové číslo permutácie), ktoré sú porovnávané s pôvodným T. V prípade že je T_i menšie ako pôvodné T, je počítadlo C_0 inkrementované o 1. V prípade rovnosti je počítadlo C_1 inkrementované o 1. Pokiaľ platí $C_0 + C_1 \leq 5$ alebo $C_0 \geq 9995$ pre ktorýkoľvek test, je predpoklad IID zamietnutý, v opačnom prípade je prijatý [6]. Kritické hodnoty C_0 a C_1 sú počítané pre pravdepodobnosť chyby prvého typu rovnej 0,001.

Permutačné testy zahŕňajú nasledujúce testy, ktoré sú popísaným spôsobom počítané nezávisle:

1. test návštev,
2. test počtu trendov,
3. test dĺžky trendov,
4. test počtu nárastov a klesaní hodnôt,
5. test počtu trendov založených na mediáne,
6. test dĺžky trendov založených na mediáne,
7. test priemeru kolízií,
8. test maximálnej kolízie,
9. test periodicity,
10. test kovariancie,
11. test kompresie.

Testy sú navrhnuté pre binárne aj nebinárne dáta. V niektorých prípadoch však neupravené binárne dáta významne ovplyvnia distribúciu dát, preto sú k dispozícii dve konverzie:

- **Konverzia I**, kedy sú binárne dáta rozdelené do blokov po 8 neprekrývajúcich sa bitov a nahradené súčtom jednotiek v každom takomto bloku. V prípade, že blok nemá dostatočnú veľkosť, je doplnený nulami. Napríklad blok dát o veľkosti 20 bitov (1,0,0,0,1,1,1,0,1,1,0,1,1,0,1,1,0,0,1,1) by po aplikovaní Konverzie I vyzeral ako (4, 6, 2).
- **Konverzia II** takisto rozdelí vstupné dáta do blokov po 8 neprekrývajúcich sa bitoch, zostaví 8-bitové číslo, ktoré je následne prevedené do desiatkovej sústavy. V prípade bloku kratšieho ako 8 bitov je rovnako doplnený nulami na konci. Pre vstupné dáta (1,0,0,0,1,1,1,0,1,1,0,1,1,0,1,1,0,0,1,1) by Konverzia II vrátila výsledok (142, 219, 48).

Test návštev

Test návštev skúma, ako sa priebežná suma vzoriek odchyľuje od priemeru v každom bode predloženého súboru. Pokiaľ vychádzame zo súboru $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, štatistický parameter T je najväčšia z odchýlok počítaných nasledovne:

1. Vypočíta sa $\bar{X}$, ktoré je aritmetickým priemerom všetkých vzoriek.
2. Pre $i = 1$ idúce do L sa vypočíta $d_i = \left|\sum_{j=1}^{i} s_j - i \cdot \bar{X}\right|$.
3. Štatistický parameter T je najväčšia z hodnôt $(d_1,\ ...,\ d_L)$.

Príklad: Nech $S = (2,\ 15,\ 4,\ 10,\ 9)$. Aritmetický priemer $\bar{X} = 8$. Pre ostatné hodnoty potom platí $d_1 = |2 - 8| = 6$, $d_2 = 1$, $d_3 = 3$, $d_4 = 1$, $\mathrm{d}_5 = 0$. Štatistický parameter $T = max(6, 1, 3, 1, 0) = 6$.

Binárne dáta: nie je potrebná žiadna konverzia.

Test počtu trendov

Tento test skúma počet narastajúcich alebo klesajúcich trendov v súbore vstupných vzoriek. Pokiaľ vychádzame zo súboru $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, štatistický parameter T je počítaný nasledovne:

1. Zostrojí sa sekvencia $S' = (s'_1,\ ...,\ s'_{L-1})$, kde s'_i je -1 v prípade, že $s_i > s'_{i+1}$ a $+1$ v prípade, že $s_i \leq s'_{i+1}$, pre $i\ = 1,\ ...,\ L - 1$.
2. Štatistický parameter T je **počet** trendov (neprerušených skupín $+1$ alebo -1) v súbore S'.

Príklad: Nech $S = (2,\ 2,\ 2,\ 5,\ 7,\ 7,\ 9,\ 3,\ 1,\ 4,\ 4)$. Potom sekvencia $S' = (+1, +1, +1, +1, +1, +1, -1, -1, +1, +1)$. To dáva tri trendy $(+1, +1, +1, +1, +1, +1)$, $(-1, -1)$, $(+1, +1)$, takže $T = 3$.

Binárne dáta: je potrebná **Konverzia I**.

Test dĺžky trendov

Tento test skúma dĺžku narastajúcich alebo klesajúcich trendov v súbore vstupných vzoriek. Pokiaľ vychádzame zo súboru $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, štatistický parameter T je počítaný nasledovne:

1. Zostrojí sa sekvencia $S' = (s'_1,\ ...,\ s'_{L-1})$, kde s'_i je -1 v prípade, že $s_i > s'_{i+1}$ a $+1$ v prípade, že $s_i \leq s'_{i+1}$, pre $i\ = 1,\ ...,\ L-1$.
2. Štatistický parameter T je **dĺžka najdlhšieho** trendu v súbore S'.

Príklad: Nech $S = (2,\ 2,\ 2,\ 5,\ 7,\ 7,\ 9,\ 3,\ 1,\ 4,\ 4)$. Potom sekvencia $S' = (+1, +1, +1, +1, +1, +1, -1, -1, +1, +1)$. To dáva tri trendy $(+1, +1, +1, +1, +1, +1)$, $(-1, -1)$, $(+1, +1)$, takže $T = 6$.

Binárne dáta: je potrebná **Konverzia I**.

Test počtu nárastov a klesaní hodnôt

Test počtu nárastov a klesaní hodnôt skúma rozdiely (nárasty a klesania) medzi po sebe nasledujúcimi vzorkami. Pokiaľ vychádzame zo súboru $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, štatistický parameter T je počítaný nasledovne:

1. Zostrojí sa sekvencia $S' = (s'_1,\ ...,\ s'_{L-1})$, kde s'_i je -1 v prípade, že $s_i > s'_{i+1}$ a $+1$ v prípade, že $s_i \leq s'_{i+1}$, pre $i\ = 1,\ ...,\ L-1$.
2. Spočíta sa počet -1 a $+1$ v súbore S'. Štatistický parameter T je väčšie z týchto čísel.

Príklad: Nech $S = (2,\ 2,\ 2,\ 5,\ 7,\ 7,\ 9,\ 3,\ 1,\ 4,\ 4)$. Potom sekvencia $S' = (+1, +1, +1, +1, +1, +1, -1, -1, +1, +1)$. Počet $+1$ je 8, počet -1 je 2, takže $T = max(8, 2) = 8$.

Binárne dáta: je potrebná **Konverzia I**.

Test počtu trendov založených na mediáne

Tento test je svojou podstatou podobný testu počtu trendov, avšak nárast alebo klesanie sa nepočíta medzi po sebe idúcimi vzorkami, ale voči mediánu. Pokiaľ vy-

chádzame zo súboru $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, štatistický parameter T je počítaný nasledovne:

1. Nájde sa medián $\tilde{X}$ súboru $S = (s_1,\ ...,\ s_L)$.
2. Zostrojí sa sekvencia $S' = (s'_1,\ ...,\ s'_{L-1})$, kde s'_i je -1 v prípade, že $s_i < \tilde{X}$ a $+1$ v prípade, že $s_i \geq \tilde{X}$, pre $i\ = 1,\ ...,\ L$.
3. Štatistický parameter T je **počet** trendov v súbore S'.

Príklad: Nech $S = (5, 15, 12, 1, 13, 9, 4)$. Medián vstupného súboru je 9. Potom sekvencia $S' = (-1, +1, +1, -1, +1, +1, -1)$. To dáva päť trendov (-1), $(+1, +1)$, (-1), $(+1, +1)$, (-1), takže $T = 5$.

Binárne dáta: za medián pri binárnych dátach bude považovaná hodnota $0,5$. Žiadna konverzia nie je potrebná.

Test dĺžky trendov založených na mediáne

Tento test je svojou podstatou podobný testu dĺžky trendov, avšak nárast alebo klesanie sa nepočíta medzi po sebe idúcimi vzorkami, ale voči mediánu. Pokiaľ vychádzame zo súboru $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, štatistický parameter T je počítaný nasledovne:

1. Nájde sa medián $\tilde{X}$ súboru $S = (s_1,\ ...,\ s_L)$.
2. Zostrojí sa sekvencia $S' = (s'_1,\ ...,\ s'_{L-1})$, kde s'_i je -1 v prípade, že $s_i < \tilde{X}$ a $+1$ v prípade, že $s_i \geq \tilde{X}$, pre $i\ = 1,\ ...,\ L$.
3. Štatistický parameter T je **dĺžka najdlhšieho** trendu v súbore S'.

Príklad: Nech $S = (5, 15, 12, 1, 13, 9, 4)$. Medián vstupného súboru je 9. Potom sekvencia $S' = (-1, +1, +1, -1, +1, +1, -1)$. To dáva päť trendov (-1), $(+1, +1)$, (-1), $(+1, +1)$, (-1). Najdlhší z nich má dĺžku 2, takže $T = 2$.

Binárne dáta: za medián pri binárnych dátach bude považovaná hodnota $0,5$. Žiadna konverzia nie je potrebná.

Test priemeru kolízií

Test priemeru kolízií skúma počet po sebe idúcich vzoriek pokiaľ nie je nájdený duplikát vzorky. Počíta sa nasledovne:

1. Nech je C zoznam počtu vzoriek, po ktorých sa našiel duplikát vo vstupnej sekvencii $S = (s_1,\ ...,\ s_{L-1})$, kde L je počet vzoriek v súbore. C je zo začiatku prázdny.
2. Nech $i = 1$.

3. Pokiaľ je $i < L$, hľadá sa také najmenšie j, pre ktoré sekvencia $(s_i, \ ..., \ s_{i+j-1})$ obsahuje dve rovnaké vzorky (pokiaľ taká sekvencia neexistuje, pokračuje sa ďalším krokom). j je pridané do zoznamu C a $i = i + j$.
4. Štatistický parameter T je potom **priemer** všetkých hodnôt v zozname C.

Príklad: Nech $S = (2, 1, 1, 2, 0, 1, 0, 1, 1, 2)$. Prvá kolízia (duplikát vzorky) sa vyskytne pre $j = 3$. Číslo 3 je pridané do zoznamu C a prvé tri vzorky sú zahodené – súbor má podobu (2, 0, 1, 0, 1, 1, 2). Ďalšia kolízia sa objaví pre $j = 4$. Číslo 4 je pridané do zoznamu C a prvé štyri vzorky sú zahodené, súbor má potom podobu (1, 1, 2). Kolízia sa objaví pre $j = 2$. Číslo 2 je pridané do zoznamu C a prvé dve vzorky sú zahodené, súbor má potom podobu (2). V tejto sekvencii sa už nenachádzajú ďalšie kolízie. Keďže $C = [3, 4, 2]$, priemer týchto hodnôt je 3, $T = 3$.

Binárne dáta: je potrebná **Konverzia II**.

Test maximálnej kolízie

Test maximálnej kolízie skúma počet po sebe idúcich vzoriek pokiaľ nie je nájdený duplikát vzorky. Počíta sa nasledovne:

1. Nech je C zoznam počtu vzoriek, po ktorých sa našiel duplikát vo vstupnej sekvencii $S = (s_1, \ ..., \ s_{L-1})$, kde L je počet vzoriek v súbore. C je zo začiatku prázdny.
2. Nech $i = 1$.
3. Pokiaľ je $i < L$, hľadá sa také najmenšie j, pre ktoré sekvencia $(s_i, \ ..., \ s_{i+j-1})$ obsahuje dve rovnaké vzorky (pokiaľ taká sekvencia neexistuje, pokračuje sa ďalším krokom). j je pridané do zoznamu C a $i = i + j$.
4. Štatistický parameter T je potom **najväčšia** zo všetkých hodnôt v zozname C.

Príklad: Nech $S = (2, 1, 1, 2, 0, 1, 0, 1, 1, 2)$. Prvá kolízia (duplikát vzorky) sa vyskytne pre $j = 3$. Číslo 3 je pridané do zoznamu C a prvé tri vzorky sú zahodené – súbor má podobu (2, 0, 1, 0, 1, 1, 2). Ďalšia kolízia sa objaví pre $j = 4$. Číslo 4 je pridané do zoznamu C a prvé štyri vzorky sú zahodené, súbor má potom podobu (1, 1, 2). Kolízia sa objaví pre $j = 2$. Číslo 2 je pridané do zoznamu C a prvé dve vzorky sú zahodené, súbor má potom podobu (2). V tejto sekvencii sa už nenachádzajú ďalšie kolízie. Keďže $C = [3, 4, 2]$, najväčšia z týchto hodnôt je 4, $T = 4$.

Binárne dáta: je potrebná **Konverzia II**.

Test periodicity

Testy periodicity sa zameriava na určenie počtu periodicít v skúmanom súbore $S = (s_1, \ ..., \ s_{L-1})$, kde L je počet vzoriek v súbore. Vstupným argumentom testu je oneskorovací parameter p (anglicky *lag parameter*), pričom musí platiť $p < L$. Test je opakovaný pre 5 rôznych hodnôt parametru p – hodnoty 1, 2, 8, 16 a 32. Štatistický parameter T sa počíta nasledovne:

1. Nech $T = 0$.
2. Pre $i = 1$ idúce do $L - p$; T je inkrementované o hodnotu 1 v prípade, že $s_i = s_{i+p}$.

Príklad: Nech $S = (2, 1, 2, 1, 0, 1, 0, 1, 1, 2)$ a $p = 2$. s_i sa rovná s_{i+p} pre 5 hodnôt i (1, 2, 4, 5 a 6), takže $T = 5$.

Binárne dáta: je potrebná **Konverzia I**.

Test kovariancie

Tento test skúma silu oneskorenej korelácie medzi vstupnými vzorkami v súbore $S = (s_1, \ ..., \ s_{L-1})$, kde L je počet vzoriek v súbore. Vstupným argumentom testu je oneskorovací parameter p, pričom musí platiť $p < L$. Test je opakovaný pre 5 rôznych hodnôt parametru p – hodnoty 1, 2, 8, 16 a 32. Štatistický parameter T sa počíta nasledovne:

1. Nech $T = 0$.
2. Pre $i = 1$ idúce do $L - p$; $T = T + (s_i \cdot s_{i+p})$.

Príklad: Nech $S = (5, 2, 6, 10, 12, 3, 1)$ a $p = 2$. T je potom počítané ako $T = (5 \cdot 6) + (2 \cdot 10) + (6 \cdot 12) + (10 \cdot 3) + (12 \cdot 1) = 164$.

Binárne dáta: je potrebná **Konverzia I**.

Test kompresie

Všeobecné kompresné algoritmy sú určené na odstraňovanie redundancie v zadaných reťazcoch na princípe nahradzovania opakujúcich sa podreťazcov. Test kompresie berie vstupný súbor $S = (s_1, \ ..., \ s_{L-1})$, kde L je počet vzoriek v súbore, pretransformuje ho do reťazca obsahujúceho jednotlivé vzorky oddelené medzerami a aplikuje naň kompresný algoritmus. Štatistický parameter T sa pri kompresnom teste počíta nasledovne:

1. Vstupný súbor je transformovaný do podoby reťazca obsahujúceho jednotlivé vzorky oddelené medzerami, napríklad ak je vstupný súbor $S = (144, 21, 139, 0, 0, 15)$, výstupom je reťazec `144 21 139 0 0 15`.
2. Na reťazec získaný v predošlom bode je použitý kompresný algoritmus, konkrétne ***bzip2*** [21].
3. Štatistický parameter T je dĺžka skomprimovaného reťazca v bajtoch.

Binárne dáta: nie je potrebná žiadna konverzia.

2.3.2 Chi-kvadrát testy

Táto časť obsahuje chi-kvadrátové testy, ktorých úlohou je testovanie nezávislosti a zhody modelu dát. Testy nezávislosti sa zameriavajú na závislosti medzi pravdepodobnosťami výskytu medzi po sebe idúcimi vzorkami v celom súbore. Testy zhody modelu skúmajú možnosť odlišností v distribúciach v desiatich podsúboroch získaných z pôvodného súboru vzoriek.

Test nezávislosti nebinárnych dát

Na začiatku je vstupný súbor vzoriek $S = (s_1, \ ..., \ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1, \ ..., \ x_k\}$, pričom A je množina všetkých správ. Test sa riadi nasledujúcimi krokmi:

1. Nájde sa podiel p_i každej možnej správy x_i v súbore S: $p_i = \frac{\text{počet } x_i \text{ v } S}{L}$. Spočíta sa očakávaný počet výskytov každej možnej dvojice (s_i, s_j) v súbore S ako $e_{i,j} = p_i p_j L/2$.
2. Zlúčia sa možné (s_i, s_j) dvojice, počínajúc od najmenšieho $e_{i,j}$ do kontajnerov tak, že očakávaná hodnota každého kontajneru je 5. Hodnota kontajneru sa počíta ako súčet hodnoty $e_{i,j}$ párov zahrnutých v kontajneri. Ak po zlúčení všetkých dvojíc stále existujú kontajnery s nižšou hodnotou ako 5, zlúčia sa dva s najmenšou hodnotou. Potom je n_{bin} počet takto vzniknutých kontajnerov.

Po zostavení kontajnerov prebieha test nasledovne:

1. Nech je o zoznam počítadiel o n_{bin} prvkoch, každý inicializovaný na hodnotu 0. Pre $j = 1$ idúce do $L - 1$:
   (a) Ak je dvojica (s_j, s_{j+1}) v kontajneri i, zvýši sa počítadlo o_i o 1.
   (b) Hodnota j sa inkrementuje o 2.
2. Štatistický parameter T je spočítaný ako $T = \sum_{i=1}^{n_{\text{bin}}} \frac{(o_i - E(Bin_i))^2}{E(Bin_i)}$, kde Bin_i je hodnota kontajneru a funkcia $E()$ predstavuje funkciu súčtu. Test zlyhá ak je hodnota T väčšia ako kritická hodnota chi-kvadrát testu s $(n_{\text{bin}} - 1) - (k - 1) =$

$n_{\text{bin}} - k$ stupňami voľnosti, keď pravdepodobnosť chyby prvého typu je 0,001. Ak je stupeň voľnosti menší ako jedna, test sa neaplikuje.

Príklad: Nech S = (2, 2, 3, 1, 3, 2, 3, 2, 1, 3, 1, 1, 2, 3, 1, 1, 2, 2, 2, 3, 3, 2, 3, 2, 3, 1, 2, 2, 3, 3, 2, 2, 2, 1, 3, 3, 3, 2, 3, 2, 1, 3, 2, 3, 1, 2, 2, 3, 1, 1, 3, 2, 3, 2, 3, 1, 2, 2, 3, 3, 2, 2, 2, 1, 3, 3, 3, 2, 3, 2, 1, 2, 2, 3, 3, 3, 2, 3, 2, 1, 2, 2, 2, 1, 3, 3, 3, 2, 3, 2, 1, 3, 2, 3, 1, 2, 2, 3, 1, 1). Súbor pozostáva z $k = 3$ správ $\{1, 2, 3\}$ s pravdepodobnosťami $p_1 = 0,21$, $p_2 = 0,41$ a $p_3 = 0,38$. S dĺžkou $L = 100$, očakávaný počet výskytov každej dvojice je možné nájsť v tabuľke 2.3.

Tab. 2.3: Vzostupne zoradené očakávané počty výskytov každej dvojice správ zdroja.

(s_i, s_j)	(1,1)	(1,3)	(3,1)	(1,2)	(2,1)	(3,3)	(2,3)	(3,2)	(2,2)
$e_{i,j}$	2,21	3,99	3,99	4,31	4,31	7,22	7,79	7,79	8,41

Dvojice môžu byť zlúčené do $n_{\text{bin}} = 6$ kontajnerov. Hodnotu jednotlivých kontajnerov je možné nájsť v tabuľke 2.4.

Tab. 2.4: Hodnoty kontajnerov.

Kontajner	Dvojica	Hodnota $E(Bin_i)$
1	(1,1), (1,3)	6,2
2	(3,1), (1,2)	8,3
3	(2,1), (3,3)	11,53
4	(2,3)	7,79
5	(3,2)	7,79
6	(2,2)	8,41

Skutočné výskyty pre jednotlivé kontajnery sú spočítané ako 7, 6, 10, 8, 12 a 7 a štatistický parameter $T = 3,46$. Stupeň voľnosti je 3 (= 6 − 3). Hypotéza nie je zamietnutá, nakoľko štatistický parameter je nižší ako kritická hodnota 16,266.

Test zhody modelu nebinárnych dát

Na začiatku je vstupný súbor vzoriek $S = (s_1, \ ..., \ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1, \ ..., \ x_k\}$, pričom A je množina všetkých správ. Test sa riadi nasledujúcimi krokmi:

1. Nech c_i je počet výskytov x_i v celom súbore S a nech $e_i = c_i/10$, pre $1 \leq i \leq k$. c_i je delené desiatimi z dôvodu delenia súboru na 10 podsúborov.

2. Nech $Zoznam[i]$ obsahuje správu s i-tym najmenším e_i (napríklad, $Zoznam[1]$ má najmenšiu hodnotu e_i, $Zoznam[2]$ má druhú najmenšiu hodnotu, atď.).
3. Začínajúc od zoznamu $Zoznam[1]$, zlúčia sa správy do kontajnerov. Do kontajnera sa priraďujú aj ďalšie zoznamy $Zoznam[i]$ až do chvíle, kým súčet e_i daného kontajneru nedosiahne aspoň hodnotu 5, následne sa položky začnú priraďovať do ďalšieho kontajneru. Pokiaľ je hodnota posledného kontajneru menšia ako 5, zlúčia sa posledné dva kontajnery. Počet takto zostavených kontajnerov je n_{bin}.
4. Nech je E_i počet očakávaných správ v kontajneri i; E_i je súčet e_i vzoriek v danom kontajneri. Napríklad, ak kontajner 1 obsahuje vzorky (x_1, x_{10}, x_{50}), $E_1 = e_1 + e_{10} + e_{50}$.

Príklad: Nech je počet možných správ $k = 4$ a $c_1 = 43$, $c_2 = 55$, $c_3 = 52$ a $c_4 = 10$. Po rozdelení celého vstupu na 10 častí, očakávaný počet výskytov každej správy je $e_1 = 4,3$, $e_2 = 5,5$, $e_3 = 5,2$ a $e_4 = 1$. Zoznam správ začínajúci najmenším očakávaným počtom výskytov je $Zoznam = [4, 1, 3, 2]$. Prvý kontajner obsahuje správy 4 a 1 a očakávaná hodnota kontajneru 1 je 5,3 ($= e_4 + e_1$). Druhý kontajner obsahuje správu 3 a posledný kontajner obsahuje správu 2. Nakoľko je hodnota posledného kontajneru väčšia ako 5, nie je potrebné žiadne zlučovanie.

S daným n_{bin}, E_i a zoznamom správ pre každý kontajner, test je vykonaný nasledovne:

1. Súbor S je rozdelený na 10 neprekrývajúcich sa podsúborov o veľkosti $\left\lfloor \frac{L}{10} \right\rfloor$ ($\lfloor ... \rfloor$ vráti celé číslo zaokrúhlené nadol), kde $S_d = (s_{d\lfloor L/10 \rfloor + 1}, ..., s_{(d+1)\lfloor L/10 \rfloor})$ pre $d = 0, ..., 9$. Pokiaľ nie je L násobok 10, zostávajúce vzorky nie sú použité.
2. $T = 0$.
3. Pre $d = 0$ idúce do 9:
   (a) Pre $i = 1$ idúce do n_{bin}:
      i. Nech o_i je celkový počet výskytov správ v kontajneri i v podsúbore S_d.
      ii. $T = T + \frac{(o_i - E_i)^2}{E_i}$.

Test zlyhá, ak je štatistický parameter T väčší ako kritická hodnota chi-kvadrátu s $9 \cdot (n_{\text{bin}} - 1)$ stupňami voľnosti, pokiaľ je pravdepodobnosť chyby prvého typu stanovená na 0,001.

Test nezávislosti binárnych dát

Tento test kontroluje predpoklad nezávislosti pre binárne dáta. Môže byť použitý chi-kvadrát test pre nezávislosť medzi susediacimi bitmi, avšak test je z dôvodu

malej množiny možných správ značne obmedzený na sile. Vhodnejšou formou je porovnávanie frekvencií m-bitových tried s ich očakávanými hodnotami, ktoré sú počítané ako pravdepodobnosti každého nasledujúceho bitu, za predpokladu, že sú vzorky nezávislé. Pokiaľ nie sú susedné bity nezávislé, očakávané pravdepodobnosti m-bitových tried odvodené z ich bitových pravdepodobností budú skreslené pre celý súbor, čo vyústi do oveľa väčšej hodnoty štatistického parametru.

Na začiatku je vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore. Dĺžka m tried je počítaná ako:

1. Nech p_0 a p_1 sú pomery výskytov 0 a 1 v súbore S (tzn. $p_0 = \frac{\text{počet 0 v } S}{L}$ a $p_1 = \frac{\text{počet 1 v } S}{L}$).
2. Nájde sa najväčšie celé číslo m, ktoré spĺňa podmienku $min(p_0, p_1)^m \left\lfloor \frac{L}{m} \right\rfloor \geq 5$. Ak je m väčšie ako 11, $m = 11$. Ak je m rovné 1, test zlyhá. Napríklad, ak $p_0 = 0,14$, $p_1 = 0,86$ a $L = 1000$, tak $m = 2$.

Test je vykonaný, ak $m \geq 2$:

1. Štatistický parameter $T = 0$.
2. Súbor S je rozdelený na neprekrývajúce sa bloky o veľkosti m bitov označené ako $B = (B_1, ..., B_{\lfloor \frac{L}{m} \rfloor})$. Ak L nie je násobkom m, ostatné bity sú zahodené.
3. Pre každú možnú m-bitovú triedu $(a_1, a_2, ..., a_m)$:
   (a) Nech o je počet výskytov vzoru $(a_1, a_2, ..., a_m)$ vo vstupe B.
   (b) Nech w je počet jednotiek v $(a_1, a_2, ..., a_m)$.
   (c) Nech $e = p_1^w (p_0)^{m-w} \left\lfloor \frac{L}{m} \right\rfloor$.
   (d) $T = T + \frac{(o-e)^2}{e}$.

Test zlyhá, ak je štatistický parameter T väčší ako kritická hodnota chi-kvadrátu s $2^m - 2$ stupňami voľnosti, pokiaľ je pravdepodobnosť chyby prvého typu stanovená na 0,001.

Test zhody modelu binárnych dát

Test zhody modelu binárnych dát skúma distribúciu počtu jednotiek v neprekrývajúcich sa intervaloch vstupných vzoriek aby rozhodol, či zostáva táto distribúcia rovnaká naprieč vstupnou sekvenciou. Ak máme vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore, potom test prebieha nasledovne:

1. Nech p je pomer výskytu 1 v celom súbore S ($p = \frac{\text{počet 1 v } S}{L}$).
2. Súbor S je rozdelený na 10 neprekrývajúcich sa podsúborov o veľkosti $\left\lfloor \frac{L}{10} \right\rfloor$, kde $S_d = (s_{d\lfloor L/10 \rfloor + 1}, ..., s_{(d+1)\lfloor L/10 \rfloor})$ pre $d = 0, ..., 9$. Pokiaľ nie je L násobok 10, zostávajúce vzorky nie sú použité.
3. Štatistický parameter $T = 0$.

4. Nech je očakávaný počet núl a jednotiek v každom podsúbore S_d definovaný ako $e_0 = (1-p)\left\lfloor \frac{L}{10} \right\rfloor$ a $e_1 = p\left\lfloor \frac{L}{10} \right\rfloor$.
5. Pre $d = 0$ idúce do 9:
   (a) Nech o_0 a o_1 sú počty 0, resp. 1, v podsúbore S_d.
   (b) $T = T + \frac{(o_0 - e_0)^2}{e_0} + \frac{(o_1 - e_1)^2}{e_1}$.

Štatistický parameter T je chi-kvadrát náhodná premenná s deviatimi stupňami voľnosti. Testy zlyhá, ak T prekročí kritickú hodnotu, ktorá je pri pravdepodobnosti chyby prvého typu 0,001, daná číslom 27,887.

Test dĺžky najdlhšieho opakujúceho sa podreťazca

Tento test overuje IID predpoklad pomocou dĺžky najdlhšieho opakujúceho sa podreťazca. Ak je táto dĺžka výrazne väčšia ako očakávaná hodnota, tak test zlyhá a vyvráti IID predpoklad. Test môže byť použitý pre binárny aj nebinárny vstup.

Na začiatku je vstupný súbor vzoriek $S = (s_1, \ ..., \ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1, \ ..., \ x_k\}$, pričom A je množina všetkých správ, test prebieha podľa nasledujúceho postupu:

1. Nájde sa pomer p_i výskytu každej možnej správy x_i: $p_i = \frac{\text{počet } x_i \text{ v } S}{L}$.
2. Vypočíta sa pravdepodobnosť kolízie $p_{\text{kol}} = \sum_{i=1}^{k} p_i^2$.
3. Nájde sa dĺžka najdlhšieho opakujúceho sa podreťazca W, t.j., nájde sa najväčšie W také, že pre aspoň jedno $i \neq j$, $s_i = s_j, s_{i+1} = s_{j+1}, ..., s_{i+W-1} = s_{j+W-1}$.
4. Počet prekrývajúcich sa podsekvencií dĺžky W v súbore S je $L - W + 1$ a počet prekrývajúcich sa dvojíc podsekvencií je $\binom{L-W+1}{2}$.
5. Nech X je binomicky distribuovaná náhodná premenná s parametrami $N = \binom{L-W+1}{2}$ a pravdepodobnosťou úspechu ${p_{kol}}^W$. Vypočíta sa pravdepodobnosť že X je väčšie alebo rovné ako 1, t.j. $Pr(X \geq 1) = 1 - Pr(X = 0) = 1 - (1 - {p_{kol}}^W)^N$.

Test zlyhá, ak je pravdepodobnosť $Pr(X \geq 1)$ menšia ako 0,001.

2.4 Odhad minimálnej entropie

Jedna z požiadaviek na zdroje entropie je schopnosť spoľahlivo generovať náhodný výstup. Aby bolo možné zaistiť dostatočné množstvo entropie pre generátory využívajúce zdroj entropie, je nutné vykonať odhad výstupnej entropie zdroja. Táto časť sa bude zaoberať metódami odhadu výstupnej entropie pre zdroj šumu a stabilizačný komponent, a to aj v prípadoch použitia neschváleného kryptografického

algoritmu. Použité metódy využívajú pre odhad isté štatistické predpoklady, ktoré nemusia všetky zdroje zaručene spĺňať, preto je nutná dôkladná znalosť zdroja šumu.

Každý odhad berie ako vstup súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Popisované odhady fungujú dobre v prípadoch, kedy je entropia jednotlivých vzoriek väčšia ako 0,1 Sh. Takisto treba pamätať na to, že niektoré odhady nie sú efektívne v prípade abecedy pozostávajúcej z viac ako 256 znakov – v takej situácii je nutné vstupnú abecedu zmenšiť.

V prípade že sa jedná zdroj o nezávislých a identicky distribuovaných dát (IID), je na odhad minimálnej entropie postačujúca prvá metóda, popísaná v časti 2.4.1 (v prípade nesprávneho predpokladu o nezávislosti a identickej distribúcii dát poskytuje táto metóda nadhodnotený odhad). V opačnom prípade, kedy dáta zo zdroja IID predpoklad nespĺňajú, je pre odhad minimálnej entropie zdroja šumu nutné použiť aj ostatné metódy. Výsledným odhadom je minimum zo všetkých získaných odhadov. Rovnaké podmienky platia aj pre odhad entropie neschváleného stabilizačného komponentu.

Zoznam odhadovacích metód, pričom metódy číslo 2, 3 a 4 sú môžu byť použité iba na binárne dáta:

1. odhad pomocou najčastejšej hodnoty,
2. odhad pomocou kolízií,
3. Markovov odhad,
4. odhad pomocou kompresie,
5. odhad pomocou t-triedy,
6. odhad pomocou najdlhšieho opakovaného podreťazca,
7. MultiMCW odhad,
8. odhad pomocou oneskorenia,
9. MultiMMC odhad,
10. LZ78Y odhad.

2.4.1 Odhad pomocou najčastejšej hodnoty

Táto metóda najprv hľadá pomer $\widehat{p}$ výskytu najčastejšie sa vyskytujúcej hodnoty v celom vstupnom súbore a potom zostrojí interval istoty pre tento pomer. Horná hranica istoty intervalu je použitá k odhadu minimálnej entropie vzoriek z tohto zdroja.

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore

a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nájde sa pomer $\widehat{p}$ výskytu najčastejšie sa vyskytujúcej hodnoty v celom vstupnom súbore, $\widehat{p} = \max_i \frac{\text{počet } x_i \text{ v } S}{L}$.
2. Vypočíta sa horná hranica intervalu istoty pravdepodobnosti výskytu najčastejšej hodnoty p_u ako $p_u = min\left(1,\ \widehat{p} + 2,576\sqrt{\frac{\widehat{p}(1-\widehat{p})}{L-1}}\right)$, kde 2,576 zodpovedá hodnote $Z_{(1-0,005)}$.
3. Odhadovaná minimálna entropia je $-log_2(p_u)$ [Sh/znak].

Príklad: Ak je súbor S = (0, 1, 1, 2, 0, 1, 2, 2, 0, 1, 0, 1, 1, 0, 2, 2, 1, 0, 2, 1), s počtom vzoriek $L = 20$, je najčastejšia hodnota 1, s $\widehat{p} = 0,4$ a $p_u = 0,4+2,576\sqrt{0,012} = 0,6895$. Odhad minimálnej entropie je $-log_2(0,6895) = 0,5363$ Sh/znak.

2.4.2 Odhad pomocou kolízií

Odhad pomocou kolízií meria stredný počet vzoriek do prvej kolízie vo vstupnom súbore, pričom pod kolíziou chápeme akúkoľvek opakovanú hodnotu. Cieľom tejto metódy je odhadnúť pravdepodobnosť najpravdepodobnejšieho výstupu, založenú na počte kolízií. Táto metóda poskytne nízky odhad entropie pre zdroje šumu, ktoré majú značné skreslenie pravdepodobnosti určitej hodnoty alebo výstupu (t.j., stredný počet vzoriek do kolízie je nízky), a vyšší odhad pre zdroje s väčším stredným počtom vzoriek do kolízie. **Tento odhad je použitý len pre binárne dáta.**

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{0, 1\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nech $v = 0$ a $index = 1$.
2. Začínajúc s s_{index}, hodnota $index$ sa zvyšuje o 1 pokiaľ sa prvá hodnota s_{index} v tomto kroku neopakuje. Inými slovami, hľadá sa najmenšie j také, že $s_i = s_j$, pričom pre i platí $index \leq i \leq j$.
3. $v = v + 1$, $t_v = j - index + 1$ a $index = j + 1$.
4. Kroky 2-3 sa opakujú pokiaľ nie je dosiahnutý koniec súboru.
5. Vypočíta sa priemer $\bar{X}$ a štandardná odchýlka $\hat{\sigma}$ premennej t_i ako $\bar{X} = \frac{1}{v}\sum_{i=1}^{v} t_i$, $\hat{\sigma} = \sqrt{\frac{1}{v-1}\sum_{i=1}^{v}(t_i - \bar{X})^2}$.
6. Vypočíta sa spodná hranica intervalu istoty pre priemer, založeného na normálnom rozdelení s úrovňou istoty 99% ako $\bar{X}' = \bar{X} - 2,576\frac{\hat{\sigma}}{\sqrt{v}}$.
7. S využitím binárneho hľadania sa nájde parameter p tak, že $\bar{X}' = pq^{-2}\left(1 + \frac{1}{2}(p^{-1} - q^{-1})\right)F(q) - pq^{-1} \cdot \frac{1}{2}(p^{-1} - q^{-1})$,

kde $q = 1 - p$, $p \geq q$, $F(1/z) = \Gamma(3, z)z^{-3}e^z$ a $\Gamma(a, b)$ je nekompletná Gamma funkcia definovaná ako $\int_b^\infty t^{a-1}e^{-t}dt$. Hranice binárneho hľadania by mali byť 1/2 a 1.

8. Pokiaľ binárne hľadanie vráti výsledok, odhadovaná minimálna entropia je $-log_2(p)$ [Sh/znak]. Ak binárne hľadanie nevráti výsledok, odhadovaná minimálna entropia je $-log_2(2) = 1$ Sh/znak.

Príklad: Nech je súbor S = (1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0). Kolízie súboru sú (1, 0, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (0, 1, 0), (1, 1), (1, 0, 0), (1, 1), (0, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (0, 1, 0), (1, 1). Po kroku 5, $v = 14$ a sekvencia $(t_1, \ ..., \ t_v)$ je (3, 3, 3, 3, 3, 2, 3, 2, 2, 3, 3, 3, 3, 2). Potom $\bar{X} = 2,7143$, $\hat{\sigma} = 0,4688$ a $\bar{X}' = 2,3915$. Riešením rovnice je $p = 0,7329$, z čoho dostaneme odhad minimálnej entropie 0,4483 Sh/znak.

2.4.3 Markovov odhad

V Markovovom procese prvého rádu, ďalšia hodnota vzorky závisí iba na poslednej pozorovanej hodnote vzorky. V Markovovom procese n-tého rádu závisí hodnota ďalšej vzorky len na posledných n pozorovaných hodnotách. Preto môže byť Markovov model použitý ako šablóna pre testovanie zdrojov so závislosťami, pretože poskytuje odhad minimálnej entropie na základe merania závislosti medzi po sebe idúcimi vzorkami zo vstupného súboru. Tento odhad je založený na množstve entropie v každej podsekvencii výstupu namiesto odhadu z celého výstupu.

Vzorky sú zbierané zo zdroja šumu ako sekvencie o dĺžke d. Z týchto dát sú následne vypočítané pravdepodobnosti pre počiatočný stav a pre prechody medzi každými dvoma stavmi. Tieto pravdepodobnosti sú následne použité na určenie najpravdepodobnejšej sekvencie o dĺžke d. Pravdepodobnosť výskytu tejto sekvencie je použitá k odhadu minimálnej entropie prítomnej vo všetkých takých sekvenciách generovaných zdrojom šumu. Je nutné poznamenať, že odhad minimálnej entropie sa vzťahuje len na sekvencie dĺžky d a nie je možné ju lineárne extrapolovať (t.j., sekvencia o dĺžke $2d$ nebude nutne mať dvojnásobnú hodnotu odhadu minimálnej entropie sekvencie dĺžky d). V niektorých prípadoch nemusí byť možné určiť typickú dĺžku d sekvencie počas testovania, preto sa v praxi (aj napriek tomu, že to matematicky nie je správne) počíta odhad minimálnej entropie na jednu vzorku (prepočítaný z odhadu pre sekvenciu dĺžky d), čo poskytuje odhad približujúci sa reálnej hodnote. **Tento odhad je použitý len pre binárne dáta.**

Je daný vstupný súbor vzoriek $S = (s_1, \ ..., \ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{0, 1\}$, pričom A je množina všetkých možných správ. Odhad sa riadi

nasledujúcim postupom:

1. Vypočíta sa pravdepodobnosť výskytu núl P_0 a jednotiek P_1 ako $P_0 = \frac{počet\ 0\ v\ S}{L}$ a $P_1 = 1 - P_0$.
2. Nech je $\boldsymbol{T}$ prechodová matica s rozmermi 2 x 2 $\begin{bmatrix} P_{0,0} & P_{0,1} \\ P_{1,0} & P_{1,1} \end{bmatrix}$, kde $P_{x,y}$ značí pravdepodobnosť prechodu zo stavu (znaku) x do y. Pravdepodobnosti v matici sú počítané nasledovne:
   $P_{0,0} = \frac{počet\ 00\ v\ S}{počet\ 00\ v\ S\ +\ počet\ 01\ v\ S}$, $P_{0,1} = \frac{počet\ 01\ v\ S}{počet\ 00\ v\ S\ +\ počet\ 01\ v\ S}$,
   $P_{1,0} = \frac{počet\ 10\ v\ S}{počet\ 10\ v\ S\ +\ počet\ 11\ v\ S}$, $P_{1,1} = \frac{počet\ 11\ v\ S}{počet\ 10\ v\ S\ +\ počet\ 11\ v\ S}$.
3. Nájde sa pravdepodobnosť výskytu najpravdepodobnejšej sekvencie o dĺžke 128 znakov podľa tabuľky 2.5:

Tab. 2.5: Tabuľka výpočtu pravdepodobností výskytov sekvencií o dĺžke 128 znakov.

Sekvencia	Pravdepodobnosť
00...0	$P_0 \cdot {P_{0,0}}^{127}$
0101...01	$P_0 \cdot {P_{0,1}}^{64} \cdot {P_{1,0}}^{63}$
011...1	$P_0 \cdot P_{0,1} \cdot {P_{1,1}}^{126}$
100...0	$P_1 \cdot P_{1,0} \cdot {P_{0,0}}^{126}$
1010...10	$P_1 \cdot {P_{1,0}}^{64} \cdot {P_{0,1}}^{63}$
11...1	$P_1 \cdot {P_{1,1}}^{127}$

4. Nech je $\hat{p}_{\max}$ maximálna hodnota pravdepodobnosti z tabuľky 2.5. Odhadovaná minimálna entropia je potom $min\,(-log_2(\hat{p}_{\max})/128,\ 1)$ [Sh/znak].

Príklad: Je daný súbor S = (1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0) s dĺžkou $L = 40$. $P_0 = 0,475$ a $P_1 = 0,525$. Prechodová matica je $\begin{bmatrix} 0,389 & 0,611 \\ 0,571 & 0,429 \end{bmatrix}$. Pravdepodobnosti možných sekvencií sú v tabuľke 2.6.

Výsledný odhad minimálnej entropie je $min\,(log_2(4,6288 \cdot 10^{-30})/128,\ 1) = min(0,761,\ 1) = 0,761$ Sh/znak.

2.4.4 Odhad pomocou kompresie

Odhad pomocou entropie odhaduje množstvo entropie na výstupu na základe toho, ako veľmi môžu byť dáta komprimované. Odhad je počítaný generovaním slovníka hodnôt a počítaním priemerného počtu vzoriek potrebného na zostavenie výstupu založeného na vytvorenom slovníku. Výhodou tohto odhadu je, že implicitne nepredpokladá nezávislosť dát. Pokiaľ je odhadovaná entropia výstupu, ktorý nie je

Tab. 2.6: Pravdepodobnosti možných sekvencií v príklade Markovovho odhadu.

Sekvencia	Pravdepodobnosť
00...0	$3,9837 \cdot 10^{-53}$
0101...01	$4,4813 \cdot 10^{-30}$
011...1	$1,4202 \cdot 10^{-47}$
100...0	$6,4631 \cdot 10^{-53}$
1010...10	$4,6288 \cdot 10^{-30}$
11...1	$1,1021 \cdot 10^{-57}$

nezávislý, kompresný pomer (a tým pádom aj výsledná odhadovaná entropia) je ovplyvnený, avšak výstupný odhad je stále udelený. Tento odhad je navyše efektívny, nakoľko stačí iba jeden priechod naprieč vstupným súborom pre vytvorenie odhadu.

Na začiatku sú vzorky vstupného súboru zo zdroja šumu rozdelené do dvoch skupín, ktoré nemajú spoločný prienik. Prvá skupina slúži ako slovník pre kompresný algoritmus, druhá skupina je použitá ako testovacia. Kompresné hodnoty sú počítané nad testovacou skupinou pre určenie priemeru. Následne je pomocou rovnakej metódy ako v **odhade pomocou kolízií** (2.4.2) vypočítané rozdelenie pravdepodobnosti, ktoré má najmenší možný odhad výstupnej entropie. Pre túto distribúciu je následne vypočítaný odhad výstupnej entropie na jednu vzorku ako spodná hranica entropie, ktorá je vo vstupnom súbore. **Tento odhad je použitý len pre binárne dáta.**

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{0, 1\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nech $b = 6$. Vytvorí sa nový súbor $S' = (s'_1,\ ...,\ s'_{\lfloor L/b \rfloor})$ rozdelením súboru S na neprekrývajúce sa bloky o dĺžke b bitov. Pokiaľ nie je L násobkom b, zvyšné vzorky sa zahodia.
2. Súbor S' sa rozdelí na dve skupiny, ktoré nemajú žiadny spoločný prienik. Tieto dve skupiny vytvoria slovník a testovacie dáta.
   (a) Vytvorí sa slovník z prvých $d = 1000$ vzoriek súboru S', $(s'_1,\ ...,\ s'_d)$.
   (b) Ostatných $v = \lfloor L/b \rfloor - d$ vzoriek, $(s'_{d+1},\ ...,\ s'_{\lfloor L/b \rfloor})$, sa použije k testovaniu.
3. Slovník *dict* sa inicializuje ako pole núl o veľkosti 2^b. Pre i idúce od 1 do d, nech $dict[s'_i] = i$. Hodnota $dict[s'_i]$ je potom index posledného výskytu s'_i v slovníku.

4. Testovacie dáta sa spracujú pomocou slovníka z bodu 2.
   (a) Nech D je zoznam s dĺžkou v.
   (b) Pre i idúce od $d+1$ do $\lfloor L/b \rfloor$:
      i. Ak je $dict[s'_i]$ nenulové, potom $D_{i-d} = i - dict[s'_i]$. Slovník sa aktualizuje indexom výskytu posledného prvku $dict[s'_i] = i$.
      ii. Ak je $dict[s'_i]$ nulové, pridá sa hodnota do slovníku $dict[s'_i] = i$. Potom $D_{i-d} = i$.
5. Vypočíta sa priemer $\bar{X}$ a štandardná odchýlka $\hat{\sigma}$ z $(log_2(D_1),\ ...,\ log_2(D_v))$, $\bar{X} = \frac{\sum_{i=1}^{v} log_2(D_i)}{v}$. Ak uvažujeme korekčný faktor $c = 0,5907$, potom $\hat{\sigma} = c\sqrt{\frac{\sum_{i=1}^{v}(log_2(D_i))^2}{v-1} - \bar{X}^2}$.
6. Vypočíta sa spodná hranica intervalu istoty pre priemer, založeného na normálnom rozdelení s úrovňou istoty 99% ako $\bar{X}' = \bar{X} - 2,576\frac{\hat{\sigma}}{\sqrt{v}}$.
7. S použitím binárneho hľadania, nájde sa parameter p tak, aby bol vzťah $\bar{X}' = G(p) + (2^b - 1)G(q)$ pravdivý, pričom $G(z) = \frac{1}{v}\sum_{t=d+1}^{\lfloor L/b \rfloor}\sum_{u=1}^{t} log_2(u)F(z,t,u)$. $F(z,t,u)$ sa pre $u < t$ rovná $z^2(1-z)^{u-1}$ a pre $u = t$ je rovné $z(1-z)^{t-1}$. Pre hodnotu q platí $q = \frac{1-p}{2^b-1}$. Hranice binárneho hľadania by mali byť 2^{-b} a 1.
8. Pokiaľ binárne hľadanie vráti výsledok, odhadovaná minimálna entropia je $-log_2(p)/b$ [Sh/znak]. Ak binárne hľadanie nevráti výsledok, odhadovaná minimálna entropia je 1 Sh/znak.

Príklad: Pre ilustráciu, nech je $d = 4$ (namiesto 1 000), súbor S = (1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1 ,1, 1, 0, 0, 0, 1, 1) s dĺžkou $L = 48$ a. Po prvom kroku bude nový súbor S' = (100011, 100101, 010111, 001100, 011100, 101010, 111011, 100011). Slovník potom bude (100011, 100101, 010111, 001100) a testovacie dáta (011100, 101010, 111011, 100011). Hodnota $v = 4$. Po inicializácii a naplnení slovníku v kroku 3 obsahuje hodnoty v tabuľke 2.7 (sú zobrazené iba nenulové hodnoty).

Tab. 2.7: Stav slovníku po naplnení v príklade odhadu pomocou kompresie.

i	1	2	3	4
s'_i	100011	100101	010111	001100
$dict[s'_i]$	1	2	3	4

Po kroku 4, $D_1 = 5$, $D_2 = 6$, $D_3 = 7$ a $D_4 = 7$. Hodnoty spočítané v kroku 5 sú $\bar{X} = 2,6304$ a $\hat{\sigma} = 0,9074$ a hodnota pre krok 6 je $\bar{X}' = 1,4617$. Hodnota p, ktorá rieši rovnicu v kroku 7 je $p = 0,5715$. Odhadovaná minimálna entropia je potom 0,1345 Sh/znak.

2.4.5 Odhad pomocou *t*-triedy

Táto metóda skúma frekvenciu výskytu t-tried (dvojíc, trojíc, atď.), ktoré sa vyskytujú vo vstupnom súbore a na jej základe počíta odhad minimálnej entropie na jednu vzorku. Frekvencia výskytu t-triedy $(r_1,\ r_2,\ ...,\ r_t)$ v súbore $S = (s_1,\ ...,\ s_L)$ je počet i takých, že $s_i = r_1, s_{i+1} = r_2,\ ...,\ s_{i+t-1} = r_t$. Triedy sa môžu prekrývať.

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nájde sa najväčšie t také, že počet výskytov najčastejšie vyskytujúcej sa t-triedy v súbore S je aspoň 35.
2. Nech $Q[i]$ obsahuje počet výskytov najčastejšie vyskytujúcej sa i-triedy v súbore S pre $i = 1, ..., t$. Napríklad, v súbore $S =$ (2, 2, 0, 1, 0, 2, 0, 1, 2, 1, 2, 0, 1, 2, 1, 0, 0, 1, 0, 0, 0), $Q[1] = max(počet\ 0,\ počet\ 1,\ počet\ 2) = počet\ 0 = 9$ a $Q[2] = 4$ je získané ako počet výskytov dvojice 01 v súbore S.
3. Pre i idúce od 1 do t, nech $P[i] = Q[i]/(L - i + 1)$. Vypočíta sa odhad maximálnej pravdepodobnosti výskytu jednotlivej vzorky ako $P_{\max}[i] = P[i]^{1/i}$. Nech $\hat{p}_{\max} = max(P_{\max}[1],\ ...,\ P_{\max}[t])$.
4. Vypočíta sa horná hranica intervalu istoty pravdepodobnosti výskytu najčastejšej hodnoty p_u ako $p_u = min\left(1,\ \widehat{p}_{\max} + 2,576\sqrt{\frac{\hat{p}_{\max}(1-\hat{p}_{\max})}{L-1}}\right)$.
5. Odhadovaná minimálna entropia je $-log_2(p_u)$ [Sh/znak].

Príklad: Pre ilustráciu, nech je hraničná hodnota v prvom bode 3 (namiesto 35), Je daný vstupný súbor $S =$ (2, 2, 0, 1, 0, 2, 0, 1, 2, 1, 2, 0, 1, 2, 1, 0, 0, 1, 0, 0, 0) s dĺžkou $L = 21$. Počet výskytov najčastejšie sa vyskytujúcej 4-triedy je 2, čo je pod hraničnou hodnotou, preto $t = 3$. V kroku 2, $Q[1] = 9$, $Q[2] = 4$ a $Q[3] = 3$. $P[1] = 0,4286$, $P[1] = 0,2$, $P[3] = 0,1579$. $P_{\max}[1] = 0,4286$, $P_{\max}[2] = 0,4472$, $P_{\max}[3] = 0,5405$ a $\hat{p}_{\max} = 0,5405$. Horná hranica 99% intervalu istoty je 0,8276. Odhad minimálnej entropie je $-log_2(0,8276) = 0,273$ Sh/znak.

2.4.6 Odhad pomocou najdlhšieho opakovaného podreťazca

Táto metóda sa zaoberá odhadom tzv. *kolíznej* entropie, ktorá reprezentuje hornú hranicu z odhadu minimálnej entropie. Je založená počte opakovaných podreťazcov (resp. tried) vo vstupnom súbore. Na rozdiel od odhadu pomocou t-triedy (2.4.5), tento odhad sa zaoberá triedami, ktoré sú príliš dlhé pre odhad pomocou t-triedy – jedná sa o komplementárny odhad.

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nájde sa najmenšie u také, že počet výskytov najčastejšie vyskytujúcej sa u-triedy v súbore S je menej ako 35.
2. Nájde sa najväčšie v také, že počet výskytov najčastejšie vyskytujúcej sa v-triedy v súbore S je aspoň 2 a počet výskytov najčastejšie vyskytujúcej sa $(v+1)$-triedy v súbore S je 1. Inými slovami, v je najdlhšia dĺžka triedy, ktorá sa opakuje. Ak je $v < u$, odhad nie je udelený.
3. Pre W idúce od u do v, vypočíta sa odhadovaná pravdepodobnosť kolízie W-triedy ako $P_W = \frac{\sum_i \binom{C_i}{2}}{\binom{L-W+1}{2}}$, kde C_i je počet výskytov i-tej unikátnej W-triedy. Vypočíta sa odhad priemernej pravdepodobnosti kolízie na jednu vzorku ako $P_{\max,W} = {P_W}^{1/W}$. Nech $\hat{p} = max(P_{\max,u},\ ...,\ P_{\max,v})$.
4. Vypočíta sa horná hranica intervalu istoty pravdepodobnosti výskytu najčastejšej hodnoty p_u ako $p_u = min\left(1,\ \widehat{p} + 2,576\sqrt{\frac{\hat{p}(1-\hat{p})}{L-1}}\right)$.
5. Odhadovaná minimálna entropia je $-log_2(p_u)$ [Sh/znak].

Príklad: Pre ilustráciu, nech je hraničná hodnota v prvom bode 3 (namiesto 35), Je daný vstupný súbor S = (2, 2, 0, 1, 0, 2, 0, 1, 2, 1, 2, 0, 1, 2, 1, 0, 0, 1, 0, 0, 0) s dĺžkou $L = 21$. V kroku 1, $u = 4$ a počet výskytov najčastejšie sa vyskytujúcej 4-triedy je 2. V kroku 2, $v = 5$. Po kroku 3, $P_4 = 0,0131$, $P_5 = 0,0074$, $P_{\max,4} = 0,3381$, $P_{\max,5} = 0,3744$ a $\hat{p} = max(0,3381,\ 0,3744) = 0,3744$. Po kroku 4, $p_u = 0,6531$. Odhad minimálnej entropie je $-log_2(0,6531) = 0,6146$ Sh/znak.

2.4.7 Odhad pomocou najčastejšej hodnoty v okne

Odhad pomocou najčastejšej hodnoty v okne využíva viacero pod-odhadov, ktorých cieľom je uhádnuť nasledujúci výstup zo zdroja, založený na posledných w výstupoch. Každý pod-odhad odhaduje hodnotu, ktorá sa najčastejšie vyskytuje v danom okne posledných w výstupov. Tento odhad si udržiava tabuľku s hodnotením, ktorá sleduje počet správnych odhadov každého pod-odhadu a využíva pod-odhad s najväčšou úspešnosťou na odhad ďalšej hodnoty. V prípade zhodných hodnotení sa odhaduje najčastejšie vyskytujúca sa hodnota, ktorá sa objavila naposledy. Tento odhad je navrhnutý pre prípady, kedy sa najčastejšie vyskytujúce hodnoty časom menia, avšak naprieč dlhou sekvenciou zostávajú relatívne stabilné.

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nech sú veľkosti okien $w_1 = 63$, $w_2 = 255$, $w_3 = 1023$, $w_4 = 4095$ a $N = L - w_1$. Ďalej, *correct* (správne hodnotenia) je pole *booleanovských* hodnôt inicializovaných na 0, o veľkosti N.
2. Nech je *scoreboard* (tabuľka s hodnotením) zoznam 4 počítadiel, všetkých inicializovaných na hodnotu 0. Nech *frequent* (frekvencia výskytov) je zoznam pozostávajúci zo 4 hodnôt, každej inicializovanú na hodnotu *Null*. Nech $winner = 1$ (víťaz).
3. Pre i idúce od $w_1 + 1$ do L:
   (a) Pre j idúce od 1 do 4:
      i. Ak $i > w_j$, nech $frequent_j$ je najčastejšie vyskytujúca sa hodnota v $(s_{i-w_j},\ s_{i-w_j+1},\ ...,\ s_{i-1})$. Pokiaľ nastane zhoda, tak sa za najčastejšie vyskytujúcu sa hodnotu vyberie tá, ktorá sa objavila naposledy.
      ii. Inak, nech $frequent_j = Null$.
   (b) Nech $prediction = frequent_{winner}$ (odhad).
   (c) Ak $prediction = s_i$, potom $correct_{i-w_1} = 1$.
   (d) Aktualizuje sa *scoreboard*. Pre j idúce od 1 do 4:
      i. Ak $frequent_j = s_i$:
         A. Nech $scoreboard_j = scoreboard_j + 1$.
         B. Ak $scoreboard_j \geq scoreboard_{winner}$, potom $winner = j$.
4. Nech C je počet jednotiek v *correct*.
5. Vypočíta sa celková presnosť odhadu ako $P_{\text{global}} = \frac{C}{N}$. Horná hranica 99% intervalu istoty, označovaná ako P'_{global} je počítaná ako:
   - $P'_{\text{global}} = 1 - 0,01^{\frac{1}{N}}$, pokiaľ $P_{\text{global}} = 0$,
   - $P'_{\text{global}} = min\left(1,\ P_{\text{global}} + 2,576\sqrt{\frac{P_{\text{global}}(1-P_{\text{global}})}{N-1}}\right)$ v ostatných prípadoch,

   kde 2,576 zodpovedá hodnote $Z_{(1-0,005)}$.
6. Vypočíta sa čiastková presnosť odhadu, založená na najdlhšej sérii správnych odhadov. Nech je r o jedna väčšie ako najdlhšia séria jednotiek v *correct*. S využitím binárneho hľadania sa nájde riešenie nasledujúcej rovnice pre P_{local}: $0,99 = \frac{1-P_{\text{local}}x}{(r+1-rx)q} \cdot \frac{1}{x^{N+1}}$, kde $q = 1 - P_{\text{local}}$ a $x = x_{10}$, odvodené z rekurentného vzťahu $x_j = 1 + qP^r_{\text{local}}x^{r+1}_{j-1}$, pre j idúce od 1 do 10 a $x_0 = 1$.
7. Výsledný odhad minimálnej entropie je záporne vzatý logaritmus z vyššej presnosti: $-log_2\left(max(P'_{\text{global}}, P_{\text{local}}, \frac{1}{k})\right)$ [Sh/znak].

Príklad: Je daný súbor S = (1, 2, 1, 0, 2, 1, 1, 2, 2, 0, 0, 0) s dĺžkou $L = 12$. Pre účely tohto príkladu, nech $w_1 = 3$, $w_2 = 5$, $w_3 = 7$, $w_4 = 9$ (namiesto $w_1 = 63$, $w_2 = 255$, $w_3 = 1023$, $w_4 = 4095$). Potom $N = 9$. Po kroku 3 sú hodnoty ako v tabuľke 2.8.

Tab. 2.8: Stav po 3. kroku v príklade odhadu pomocou najčastejšej hodnoty v okne.

i	*frequent* (krok 3b)	*scoreboard* (krok 3b)	*winner*	*prediction*	s_i	$correct_{i-w_1}$	*scoreboard* (krok 3d)
4	(1, –, –, –)	(0, 0, 0, 0)	1	1	0	0	(0, 0, 0, 0)
5	(0, –, –, –)	(0, 0, 0, 0)	1	0	2	0	(0, 0, 0, 0)
6	(2, 2, –, –)	(0, 0, 0, 0)	1	2	1	0	(0, 0, 0, 0)
7	(1, 1, –, –)	(0, 0, 0, 0)	1	1	1	1	(1, 1, 0, 0)
8	(1, 1, 1, –)	(1, 1, 0, 0)	2	1	2	0	(1, 1, 0, 0)
9	(1, 2, 2, –)	(1, 1, 0, 0)	2	2	2	1	(1, 2, 1, 0)
10	(2, 2, 2, 2)	(1, 2, 1, 0)	2	2	0	0	(1, 2, 1, 0)
11	(2, 2, 2, 2)	(1, 2, 1, 0)	2	2	0	0	(1, 2, 1, 0)
12	(0, 0, 2, 0)	(1, 2, 1, 0)	2	0	0	1	(2, 3, 1, 1)

Potom, ako sú vykonané všetky odhady, *correct* = (0, 0, 0, 1, 0, 1, 0, 0, 1). Následne, $P_{\text{global}} = 0,3333$, $P'_{\text{global}} = 0,7627$, $P_{\text{local}} = 0,036$ a výsledná odhadovaná minimálna entropia je 0,3908 Sh/znak.

2.4.8 Odhad pomocou oneskorenia

Tento odhad takisto obsahuje niekoľko pod-odhadov, ktorých cieľom je uhádnuť nasledujúci výstup zdroja na základe špecifikovaného oneskorenia. Odhad pomocou oneskorenia udržiava tabuľku s hodnotením, ktorá zaznamenáva počet správnych odhadov každého pod-ohadu a na základe tohto hodnotenia vyberá pod-odhad, ktorý bude odhadovať nasledujúci výstup zdroja.

Je daný vstupný súbor vzoriek $S = (s_1, \ ..., \ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1, \ ..., \ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nech $D = 128$ a $N = L - 1$. Ďalej, *lag* (oneskorenie) je zoznam pozostávajúci z D hodnôt, každej inicializovanú na hodnotu *Null*. Nech je *correct* (správne hodnotenia) pole *booleanovských* hodnôt inicializovaných na 0, o veľkosti N.
2. Nech je *scoreboard* (tabuľka s hodnotením) zoznam D počítadiel, všetkých inicializovaných na hodnotu 0. Nech *winner* = 1 (víťaz).
3. Pre i idúce od 2 do L:
   (a) Pre d idúce od 1 do D:
      i. Ak $d < i$, $lag_d = s_{i-d}$.
      ii. Inak, nech $lag_d = Null$.

(b) Nech $prediction = lag_{winner}$ (odhad).
(c) Ak $prediction = s_i$, potom $correct_{i-1} = 1$.
(d) Aktualizuje sa *scoreboard*. Pre d idúce od 1 do D:
i. Ak $lag_d = s_i$:
A. Nech $scoreboard_d = scoreboard_d + 1$.
B. Ak $scoreboard_d \geq scoreboard_{winner}$, potom $winner = d$.

4. Nech C je počet jednotiek v *correct*.
5. Vypočíta sa celková presnosť odhadu ako $P_{\text{global}} = \frac{C}{N}$. Horná hranica 99% intervalu istoty, označovaná ako P'_{global} je počítaná ako:
   - $P'_{\text{global}} = 1 - 0,01^{\frac{1}{N}}$, pokiaľ $P_{\text{global}} = 0$,
   - $P'_{\text{global}} = min\left(1,\ P_{\text{global}} + 2,576\sqrt{\frac{P_{\text{global}}(1-P_{\text{global}})}{N-1}}\right)$ v ostatných prípadoch,

   kde 2,576 zodpovedá hodnote $Z_{(1-0,005)}$.
6. Vypočíta sa čiastková presnosť odhadu, založená na najdlhšej sérii správnych odhadov. Nech je r o jedna väčšie ako najdlhšia séria jednotiek v *correct*. S využitím binárneho hľadania sa nájde riešenie nasledujúcej rovnice pre P_{local}: $0,99 = \frac{1-P_{\text{local}}x}{(r+1-rx)q} \cdot \frac{1}{x^{N+1}}$, kde $q = 1 - P_{\text{local}}$ a $x = x_{10}$, odvodené z rekurentného vzťahu $x_j = 1 + qP^r_{\text{local}}x^{r+1}_{j-1}$, pre j idúce od 1 do 10 a $x_0 = 1$.
7. Výsledný odhad minimálnej entropie je záporne vzatý logaritmus z vyššej presnosti: $-log_2\left(max(P'_{\text{global}}, P_{\text{local}}, \frac{1}{k})\right)$ [Sh/znak].

Príklad: Je daný súbor S = (2, 1, 3, 2, 1, 3, 1, 3, 1, 2) s dĺžkou $L = 12$. Pre účely tohto príkladu, nech $D = 3$ (namiesto 128). Potom $N = 9$. Po kroku 3 sú hodnoty ako v tabuľke 2.9.

Potom, ako sú vykonané všetky odhady, *correct* = (0, 0, 0, 1, 1, 0, 0, 0, 0). Následne, $P_{\text{global}} = 0,2222$, $P'_{\text{global}} = 0,6008$, $P_{\text{local}} = 0,1167$ a výsledná odhadovaná minimálna entropia je 0,735 Sh/znak.

2.4.9 Markovov odhad s počítaním

Markovov odhad s počítaním pozostáva z niekoľkých pod-ohadov založených na Markovovom modeli s počítaním. Na rozdiel od jednoduchého Markovovho odhadu (2.4.3, ktorý počíta pravdepodobnosti prechodov), tento odhad zaznamenáva frekvencie prechodu z jedného výstupu na ďalší výstup a vytvára odhad na základe najčastejšie pozorovaných prechodov z aktuálneho výstupu. Obsahuje D paralelne bežiacich pod-odhadov, jeden pre každú hĺbku od 1 do D. Napríklad, odhad s hĺbkou 1 vytvorí model prvého rádu a odhad s hĺbkou D vytvorí model D-teho rádu (ako v 2.4.3). Tento odhad udržiava tabuľku s hodnotením, ktorá zaznamenáva počet správnych odhadov každého pod-odhadu a na základe tohto hodnotenia vyberá

Tab. 2.9: Stav po 3. kroku v príklade odhadu pomocou oneskorenia.

i	*lag*	*winner* (krok 3b)	*prediction*	s_i	$correct_{i-1}$	*scoreboard* (krok 3d)
2	(2, –, –)	1	2	1	0	(0, 0, 0)
3	(1, 2, –)	1	1	3	0	(0, 0, 0)
4	(3, 1, 2)	1	3	2	0	(0, 0, 1)
5	(2, 3, 1)	3	1	1	1	(0, 0, 2)
6	(1, 2, 3)	3	3	3	1	(0, 0, 3)
7	(3, 1, 2)	3	2	1	0	(0, 1, 3)
8	(1, 3, 1)	3	1	3	0	(0, 2, 3)
9	(3, 1, 3)	3	3	1	0	(0, 3, 3)
10	(1, 3, 1)	2	3	2	0	(0, 3, 3)

pod-odhad, ktorý bude odhadovať nasledujúci výstup zdroja.

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nech $D = 16$ a $N = L - 2$. Ďalej, *subpredict* (pod-odhad) je zoznam pozostávajúci z D hodnôt, každej inicializovanú na hodnotu *Null*. Nech je *correct* (správne hodnotenia) pole *booleanovských* hodnôt inicializovaných na 0, o veľkosti N. Nech je *entries* (položky) pole hodnôt inicializovaných na 0 a maximálny počet položiek $maxEntries = 100\ 000$.
2. Pre d idúce od 1 do D, nech je M_d skupina počítadiel, kde $M_d[x, y]$ reprezentuje počet pozorovaných prechodov z výstupu x na výstup y pre model d-teho rádu.
3. Nech je *scoreboard* (tabuľka s hodnotením) zoznam D počítadiel, všetkých inicializovaných na hodnotu 0. Nech $winner = 1$ (víťaz).
4. Pre i idúce od 3 do L:
   (a) Pre d idúce od 1 do D:
      i. Ak $d < i - 1$:
         A. Ak je $[(s_{i-d-1},\ ...,\ s_{i-2}),\ s_{i-1}]$ v M_d, zvýši sa $M_d[(s_{i-d-1},\ ...,\ s_{i-2}),\ s_{i-1}]$ o 1.
         B. Inak, ak $entries_d < maxEntries$, pridá sa počítadlo pre $[(s_{i-d-1},\ ...,\ s_{i-2}),\ s_{i-1}]$ do skupiny, $M_d[(s_{i-d-1},\ ...,\ s_{i-2}),\ s_{i-1}] = 1$ a $entries_d$ sa zvýši o 1.
   (b) Pre d idúce od 1 do D:

i. Ak $d < i$, nájde sa hodnota y, ktorá zodpovedá najvyššiemu $M_d[(s_{i-d},\ ...,\ s_{i-1}),\ y]$ a označí sa ako *ymax*. Pokiaľ nastane zhoda, bude *ymax* najvyššie y z týchto zhôd. Nech $subpredict_d = ymax$ (pododhad). Ak sú všetky možné hodnoty $M_d[(s_{i-d},\ ...,\ s_{i-1}),\ y]$ rovné 0, potom $subpredict_d = Null$.

(c) Nech $prediction = subpredict_{winner}$ (odhad).

(d) Ak $prediction = s_i$, potom $correct_{i-2} = 1$.

(e) Aktualizuje sa *scoreboard*. Pre d idúce od 1 do D:

i. Ak $subpredict_d = s_i$:

A. Nech $scoreboard_d = scoreboard_d + 1$.

B. Ak $scoreboard_d \geq scoreboard_{winner}$, potom $winner = d$.

5. Nech C je počet jednotiek v *correct*.
6. Vypočíta sa celková presnosť odhadu ako $P_{\text{global}} = \frac{C}{N}$. Horná hranica 99% intervalu istoty, označovaná ako P'_{global} je počítaná ako:
   - $P'_{\text{global}} = 1 - 0,01^{\frac{1}{N}}$, pokiaľ $P_{\text{global}} = 0$,
   - $P'_{\text{global}} = min\left(1,\ P_{\text{global}} + 2,576\sqrt{\frac{P_{\text{global}}(1-P_{\text{global}})}{N-1}}\right)$ v ostatných prípadoch,

   kde 2,576 zodpovedá hodnote $Z_{(1-0,005)}$.
7. Vypočíta sa čiastková presnosť odhadu, založená na najdlhšej sérii správnych odhadov. Nech je r o jedna väčšie ako najdlhšia séria jednotiek v *correct*. S využitím binárneho hľadania sa nájde riešenie nasledujúcej rovnice pre P_{local}: $0,99 = \frac{1-P_{\text{local}}x}{(r+1-rx)q} \cdot \frac{1}{x^{N+1}}$, kde $q = 1 - P_{\text{local}}$ a $x = x_{10}$, odvodené z rekurentného vzťahu $x_j = 1 + qP^r_{\text{local}}x^{r+1}_{j-1}$, pre j idúce od 1 do 10 a $x_0 = 1$.
8. Výsledný odhad minimálnej entropie je záporne vzatý logaritmus z vyššej presnosti: $-log_2\left(max(P'_{\text{global}}, P_{\text{local}}, \frac{1}{k})\right)$ [Sh/znak].

Príklad: Je daný súbor S = (2, 1, 3, 2, 1, 3, 1, 3, 1), s dĺžkou $L = 9$. Pre účely tohto príkladu, nech $D = 3$ (namiesto 16). Potom $N = 7$. Po každej iterácii v kroku 4 sú hodnoty ako v tabuľke 2.10.

Nech $\{x \to y : c\}$ označuje nenulový počet prechodov c zo stavu x do stavu y. Modely M_1, M_2 a M_3 po kroku 4a (krok aktualizácie modelu) sú zobrazené v tabuľke 2.11 pre každú hodnotu i.

Potom, ako sú vykonané všetky odhady, *correct* = (0, 0, 1, 1, 0, 1, 0). Následne, $P_{\text{global}} = 0,4286$, $P'_{\text{global}} = 0,9490$, $P_{\text{local}} = 0,1307$ a výsledná odhadovaná minimálna entropia je 0,0755 Sh/znak.

Tab. 2.10: Stav po 4. kroku v príklade Markovovho odhadu s počítaním.

i	*subpredict*	*scoreboard* (krok 4c)	*winner* (krok 4c)	*prediction*	s_i	$correct_{i-2}$	*scoreboard* (krok 3d)
3	(–, –, –)	(0, 0, 0)	1	*Null*	3	0	(0, 0, 0)
4	(–, –, –)	(0, 0, 0)	1	*Null*	2	0	(0, 0, 0)
5	(1, –, –)	(0, 0, 0)	1	1	1	1	(1, 0, 0)
6	(3, 3, –)	(1, 0, 0)	1	3	3	1	(2, 1, 0)
7	(2, 2, 2)	(2, 1, 0)	1	2	1	0	(2, 1, 0)
8	(3, –, –)	(2, 1, 0)	1	3	3	1	(3, 1, 0)
9	(2, 2, –)	(3, 1, 0)	1	2	1	0	(3, 1, 0)

2.4.10 LZ78Y odhad

Tento odhad je založený na LZ78 kódovaní [22] s *Bernsteinovou* Yabba schémou [23] pre pridávanie slov do slovníka. LZ78Y odhad si udržiava slová v slovníku a pokračuje v pridávaní ďalších, pokiaľ nie je dosiahnutá maximálna kapacita slovníku. Vždy keď je spracovaná vzorka z výstupu zdroja šumu, každý podreťazec v posledných B vzorkách aktualizuje alebo je pridaný do slovníku.

Je daný vstupný súbor vzoriek $S = (s_1,\ ...,\ s_L)$, kde L je počet vzoriek v súbore a kde $s_i \in A = \{x_1,\ ...,\ x_k\}$, pričom A je množina všetkých možných správ. Odhad sa riadi nasledujúcim postupom:

1. Nech $B = 16$ a $N = L - B - 1$. Nech je *correct* (správne hodnotenia) pole *booleanovských* hodnôt inicializovaných na 0, o veľkosti N. Nech je maximálny počet položiek v slovníku $maxDictionarySize = 65\ 536$.
2. Nech je D prázdny slovník a aktuálna veľkosť slovníku $dictionarySize = 0$.
3. Pre i idúce od $B + 2$ do L:
   (a) Pre j idúce od B dolu k 1:
      i. Ak sa $(s_{i-j-1},\ ...,\ s_{i-2})$ nenachádza v slovníku D a platí že $dictionarySize < maxDictionarySize$:
         A. $D[s_{i-j-1},\ ...,\ s_{i-2}]$ bude pridané do slovníku.
         B. Nech $D[s_{i-j-1},\ ...,\ s_{i-2}][s_{i-1}] = 1$.
         C. $dictionarySize = dictionarySize + 1$.
      ii. Ak sa $(s_{i-j-1},\ ...,\ s_{i-2})$ nachádza v slovníku:
         A. Nech $D[s_{i-j-1},\ ...,\ s_{i-2}][s_{i-1}] = D[s_{i-j-1},\ ...,\ s_{i-2}][s_{i-1}] + 1$.
   (b) Použije sa slovník pre odhad ďalšej hodnoty s_i. Nech je odhad $prediction = Null$ a nech maximálny počet $maxcount = 0$. Pre j idúce od B dolu k 1:

Tab. 2.11: Stav modelov po kroku 4a v príklade Markovovho odhadu s počítaním.

i	M_1	M_2	M_3
3	$\{2 \to 1 : 1\}$	–	–
4	$\{1 \to 3 : 1\}$, $\{2 \to 1 : 1\}$	$\{(2,1) \to 3 : 1\}$	–
5	$\{1 \to 3 : 1\}$, $\{2 \to 1 : 1\}$, $\{3 \to 2 : 1\}$	$\{(1,3) \to 2 : 1\}$, $\{(2,1) \to 3 : 1\}$	$\{(2,1,3) \to 2 : 1\}$
6	$\{1 \to 3 : 1\}$, $\{2 \to 1 : 2\}$, $\{3 \to 2 : 1\}$	$\{(1,3) \to 2 : 1\}$, $\{(2,1) \to 3 : 1\}$, $\{(3,2) \to 1 : 1\}$	$\{(1,3,2) \to 1 : 1\}$, $\{(2,1,3) \to 2 : 1\}$
7	$\{1 \to 3 : 2\}$, $\{2 \to 1 : 2\}$, $\{3 \to 2 : 1\}$	$\{(1,3) \to 2 : 1\}$, $\{(2,1) \to 3 : 2\}$, $\{(3,2) \to 1 : 1\}$	$\{(1,3,2) \to 1 : 1\}$, $\{(2,1,3) \to 2 : 1\}$, $\{(3,2,1) \to 3 : 1\}$
8	$\{1 \to 3 : 2\}$, $\{2 \to 1 : 2\}$, $\{3 \to 1 : 1\}$, $\{3 \to 2 : 1\}$	$\{(1,3) \to 1 : 1\}$, $\{(1,3) \to 2 : 1\}$, $\{(2,1) \to 3 : 2\}$, $\{(3,2) \to 1 : 1\}$	$\{(1,3,2) \to 1 : 1\}$, $\{(2,1,3) \to 1 : 1\}$, $\{(2,1,3) \to 2 : 1\}$, $\{(3,2,1) \to 3 : 1\}$
9	$\{1 \to 3 : 3\}$, $\{2 \to 1 : 2\}$, $\{3 \to 1 : 1\}$, $\{3 \to 2 : 1\}$	$\{(1,3) \to 1 : 1\}$, $\{(1,3) \to 2 : 1\}$, $\{(2,1) \to 3 : 2\}$, $\{(3,1) \to 3 : 1\}$, $\{(3,2) \to 1 : 1\}$	$\{(1,3,1) \to 3 : 1\}$, $\{(1,3,2) \to 1 : 1\}$, $\{(2,1,3) \to 1 : 1\}$, $\{(2,1,3) \to 2 : 1\}$, $\{(3,2,1) \to 3 : 1\}$

i. Nech je predošlé slovo $prev = (s_{i-j},\ ...,\ s_{i-1})$.

ii. Ak sa $prev$ nachádza v slovníku, nájde sa $y \in \{x_1,\ ...,\ x_k\}$, ktoré má najvyššiu hodnotu $D[prev][y]$. V prípade zhody je vybraté také y, ktoré má vyššiu bitovú hodnotu. Napríklad, ak majú $D[prev][1]$ a $D[prev][5]$ najvyššiu hodnotu, tak $y = 5$.

iii. Ak $D[prev][y] > maxcount$:

A. $prediction = y$.

B. $maxcount = D[prev][y]$.

(c) Ak $prediction = s_i$, nech $correct_{i-B-1} = 1$.

4. Vypočíta sa celková presnosť odhadu ako $P_{\text{global}} = \frac{C}{N}$. Horná hranica 99% intervalu istoty, označovaná ako P'_{global} je počítaná ako:

- $P'_{\text{global}} = 1 - 0,01^{\frac{1}{N}}$, pokiaľ $P_{\text{global}} = 0$,

- $P'_{\text{global}} = min\left(1,\ P_{\text{global}} + 2,576\sqrt{\frac{P_{\text{global}}(1-P_{\text{global}})}{N-1}}\right)$ v ostatných prípadoch,

kde 2,576 zodpovedá hodnote $Z_{(1-0,005)}$.

5. Vypočíta sa čiastková presnosť odhadu, založená na najdlhšej sérii správnych odhadov. Nech je r o jedna väčšie ako najdlhšia séria jednotiek v *correct*. S využitím binárneho hľadania sa nájde riešenie nasledujúcej rovnice pre P_{local}: $0,99 = \frac{1-P_{\text{local}}x}{(r+1-rx)q} \cdot \frac{1}{x^{N+1}}$, kde $q = 1 - P_{\text{local}}$ a $x = x_{10}$, odvodené z rekurentného vzťahu $x_j = 1 + qP^r_{\text{local}}x^{r+1}_{j-1}$, pre j idúce od 1 do 10 a $x_0 = 1$.
6. Výsledný odhad minimálnej entropie je záporne vzatý logaritmus z vyššej presnosti: $-log_2\left(max(P'_{\text{global}}, P_{\text{local}}, \frac{1}{k})\right)$ [Sh/znak].

Príklad: Je daný súbor S = (2, 1, 3, 2, 1, 3, 1, 3, 1, 2, 1, 3, 2), s dĺžkou $L = 13$. Pre účely tohto príkladu, nech $B = 4$ (namiesto 16). Potom $N = 7$. Po každej iterácii v kroku 3 sú hodnoty ako v tabuľke 2.12.

Potom, ako sú vykonané všetky odhady, *correct* = (0, 0, 1, 1, 0, 1, 1, 0). Následne, $P_{\text{global}} = 0,5$, $P'_{\text{global}} = 0,9868$, $P_{\text{local}} = 0,1229$ a výsledná odhadovaná minimálna entropia je 0,0191 Sh/znak.

Tab. 2.12: Stav po 3. kroku v príklade LZ78Y odhad.

i	Pridané do D	*prev*	Najväčší $D[prev]$ záznam	*prediction*	s_i	$correct_{i-B-1}$
6	$D[2,1,3,2][1]$	(1, 3, 2, 1)	*Null*	*Null*	3	0
	$D[1,3,2][1]$	(3, 2, 1)	*Null*			
	$D[3,2][1]$	(2, 1)	*Null*			
	$D[2][1]$	(1)	*Null*			
7	$D[1,3,2,1][3]$	(3, 2, 1, 3)	*Null*	*Null*	1	0
	$D[3,2,1][3]$	(2, 1, 3)	*Null*			
	$D[2,1][3]$	(1, 3)	*Null*			
	$D[1][3]$	(3)	*Null*			
8	$D[3,2,1,3][1]$	(2, 1, 3, 1)	*Null*	3	3	1
	$D[2,1,3][1]$	(1, 3, 1)	*Null*			
	$D[1,3][1]$	(3, 1)	*Null*			
	$D[3][1]$	(1)	3			
9	$D[2,1,3,1][3]$	(1, 3, 1, 3)	*Null*	1	1	1
	$D[1,3,1][3]$	(3, 1, 3)	*Null*			
	$D[3,1][3]$	(1, 3)	1			
	$D[1][3]$	(3)	1			
10	$D[1,3,1,3][1]$	(3, 1, 3, 1)	*Null*	3	2	0
	$D[3,1,3][1]$	(1, 3, 1)	3			
	$D[1,3][1]$	(3, 1)	3			
	$D[3][1]$	(3)	3			
11	$D[3,1,3,1][2]$	(1, 3, 1, 2)	*Null*	1	1	1
	$D[1,3,1][2]$	(3, 1, 2)	*Null*			
	$D[3,1][2]$	(1, 2)	*Null*			
	$D[1][2]$	(2)	1			
12	$D[1,3,1,2][1]$	(3, 1, 2, 1)	*Null*	3	3	1
	$D[3,1,2][1]$	(1, 2, 1)	*Null*			
	$D[1,2][1]$	(2, 1)	3			
	$D[2][1]$	(1)	3			
13	$D[3,1,2,1][3]$	(1, 2, 1, 3)	*Null*	1	2	0
	$D[1,2,1][3]$	(2, 1, 3)	1			
	$D[2,1][3]$	(1, 3)	1			
	$D[1][3]$	(3)	1			

3 Programy pre získavanie a hodnotenie entropie

Pre overovanie spoľahlivosti a odhadnutie výstupnej entropie zdroja je definované množstvo testov, ktoré dosahujú značnej komplexnosti. Tu prichádza vhod softvérové riešenie zahŕňajúce implementáciu postupov popísaných v kapitole 2. Táto časť práce sa bude zaoberať popisom realizácie dvoch programov – **programu pre získavanie bitových postupností zo zdrojov entropie** a **programu pre hodnotenie zdrojov entropie**.

Dôležitou časťou cyklu vývoja softvéru je analýza požiadaviek a navrhnutie vhodného riešenia. To zahŕňa výber programovacieho jazyka, návrh architektúry a dátových modelov, definície vstupných a výstupných rozhraní (angl. *interface*). Po prvotnom návrhu prichádza na rad programové riešenie, ktoré by sa malo držať zásad, ako sú napríklad princípy znovupoužitia kódu, komentáre na vhodných miestach a vývoj s ohľadom na budúce rozšírenia.

3.1 Program pre získavanie bitových postupností zo zdrojov entropie

V rámci práce bolo v kapitole č. 1 popísaných niekoľko zdrojov entropie (resp. zdrojov šumu) – princípy ich fungovania, parametre, spoľahlivosť a možné riziká spojené s ich využívaním. Nakoľko sú zdroje entropie priamo viazané na prácu s hardvérom, je virtualizácia a s tým súvisiace oddelenie programu od hardvérovej platformy nežiadúce.

Program je napísaný v jazyku **C/C++** [27, 28] vo verzii **14**, ku ktorému je dostupných množstvo knižníc s otvoreným kódom (angl. *open source*) umožňujúcich priamy prístup a prácu s hardvérom. Primárnym vývojovým operačným systém bola linuxová distribúcia **Ubuntu** [29] vo verzii **18.04.4 LTS** (verzia s dlhou podporou – *Long-Term Support*), na ktorej bola zároveň celá funkcionalita testovaná.

V rámci riešenia je dostupné získavanie bitových postupností z nasledujúcich zdrojov:

1. **zvuková karta** (analyzovaná v časti 1.3.8),
2. **sieťová karta** (časť 1.3.2), pričom sú dostupné dva typy dát:
   (a) surové dáta zo sieťovej prevádzky,
   (b) mikrosekundová časť časového razítka prichádzajúcich paketov,
3. **prevádzka na USB (univerzálna sériová zbernica – *Universal Serial Bus*) portoch** (zahŕňa **klávesnicu a myš** popísané v časti 1.3.1), pričom sú

takisto dostupné dva typy dát:
   (a) surové dáta z prevádzky,
   (b) mikrosekundová časť časového razítka prichádzajúcich dát,
4. **obrazový výstup** (analyzovaný v časti 1.3.7).

Počas vývoja boli analyzované aj zdroje popísané v častiach 1.3.5 (tepelný šum), 1.3.4 (systémové parametre a premenné) a 1.3.6 (otáčky magnetického disku a ventilátoru), avšak boli ako zdroje entropie zamietnuté, nakoľko k zmene výstupných hodnôt dochádzalo len veľmi pomaly alebo vôbec.

3.1.1 Zvuková karta

Zbieranie dát zo zvukovej karty je riešené prostredníctvom multiplatformovej *open-source* knižnice **PortAudio** [30]. Knižnica poskytuje jednoduché API (rozhranie pre programovanie aplikácií – *Application Programming Interface*) pre prácu so zvukovým zariadeniami. V rámci programu je vykonávaný jednoduchý záznam zvuku z predvoleného systémového vstupného zariadenia (vo väčšine prípadov mikrofón). Knižnica umožňuje zaznamenávanie zvuku v niekoľkých formátoch:

- `float32`,
- `int32`,
- `int16`,
- `int8`,
- `uint8`,

pričom každý z nich ukladá jeden segment záznamu v inej podobe a inej veľkosti. Jednotlivému porovnaniu bude venovaný priestor v nasledujúcej kapitole.

Súbory samotnej knižnice sa nachádzajú v samostatnom priečinku priloženom k finálnemu súboru (viac informácií o súborovom rozdelení prílohy je možné nájsť v prílohe A). Okrem súborov knižnice je potrebné mať v systéme nainštalované súčasti `flex`, `bison` a `libasound-dev`, ktoré umožňujú prístup k hardvéru počítača. Ich inštaláciu je možné vykonať prostredníctvom príkazu:

```
sudo apt-get install flex bison libasound-dev
```

Po nainštalovaní potrebných súčastí je nutné nainštalovať aj samotnú knižnicu. Inštaláciu je možné vykonať otvorením terminálu v priečinku s knižnicou a zadaním nasledujúcich príkazov:

```
./configure && make
sudo make install
```

Po vykonaní príkazov je možné knižnicu začať používať. Samotné riešenie nahrávania zvuku je následne veľmi jednoduché a je realizované jedinou funkciou:

```
int audio(const char* fileName, int bitsCount);
```

ktorá má dva parametre, názov súboru do ktorého je uložený výstup a počet bitov (presnejšie znakov) na uloženie. Návratovou hodnotou je kód prípadnej chyby. Užívateľ je o priebehu nahrávania pravidelne informovaný.

3.1.2 Sieťová karta a USB

Zachytávanie sieťovej prevádzky a komunikácie na USB portoch je takisto riešené prostredníctvom multiplatformovej *open-source* knižnice s názvom **libpcap** [31]. Knižnica poskytuje API pre zachytávanie komunikácie, ktoré spracúva samotný obsah komunikácie, rovnako ako aj čas jej príchodu. Pre prácu s knižnicou je nutné nainštalovať súčasť `libpcap-dev`, ktorá v systéme umožňuje zachytávať a spracúvať komunikáciu. Nainštalovať ju je možné prostredníctvom príkazu:

```
sudo apt-get install libpcap-dev
```

Je dôležité poznamenať, že väčšina moderných operačných systémov nedovoľuje zachytávanie sieťovej komunikácie bežným užívateľom, preto je nutné výsledný program spustiť s administrátorskými právami (`sudo`). Aby bolo možné odchytávať komunikáciu aj z USB portov, je nutné aktivovať systémový modul `usbmon` príkazom:

```
sudo modprobe usbmon
```

Samotná implementácie pozostáva z jednej hlavnej funkcie:

```
int network(const char *fileName,
            int bitsCount,
            bool captureTime);
```

ktorá ako v prípade zvukovej karty má prvé dva parametre názov súboru do ktorého je uložený výstup a počet bitov (presnejšie znakov) na uloženie. Tretí parameter reprezentuje nastavenie, ktoré mení správanie zaznamenávania zo zachytávania dátového obsahu paketov na zaznamenávanie mikrosekundovej časti časového razítka príchodu dát. Návratovou hodnotou je kód prípadnej chyby. Užívateľ je o priebehu nahrávania pravidelne informovaný.

3.1.3 Obrazový výstup

Vytváranie snímok obrazovky (tzv. *screenshot*) je vykonávané prostredníctvom *open-source* knižnice **X11** [32], ktorú je v prípade absencie v systéme možné nainštalovať príkazom:

```
sudo apt-get install libx11-dev
```

Táto súčasť poskytuje API, ktoré umožňuje zachytiť aktuálny obsah obrazovky, ktorý je následne uložený do súboru. Pre zachovanie konzistencie a prehľadnosti je celá funkcionalita zabalená do jedinej funkcie:

```
int screen(const char *fileName, int bitsCount);
```

ktorá má dva parametre, názov súboru do ktorého je uložený aktuálny obsah obrazovky v binárnej podobe a počet bitov (presnejšie znakov) na uloženie. Návratovou hodnotou je kód prípadnej chyby. Užívateľ je o výsledku operácie informovaný výpisom.

3.1.4 Spustenie programu

Hlavným súborom programu je `main.cpp`, ktorý obsahuje logiku načítavania vstupov, prevolávania jednotlivých funkcií a zabezpečuje celé riadenie programu. Projekt je zostavený prostredníctvom nástroja **CMake** [33], ktorý umožňuje popísať všetky závislosti programu a spôsob kompilácie v jedinom súbore s názvom `CMakeLists.txt`. Na úvodných riadkoch zobrazených vo výpise 3.1 je definovaná verzia programu **CMake** na **3.10**, názov projektu a definícia požadovanej verzie prekladača. Na nasledujúcich riadkoch sú pridávané požadované knižnice, ktoré sa v čase prekladu pripoja k hlavnému súboru.

Výpis 3.1: Základné definície zo súboru `CMakeLists.txt`.

```
1 cmake_minimum_required(VERSION 3.10)
2 project(entropy_sources)
3
4 set(CMAKE_CXX_STANDARD 14)
```

Aby bolo možné program používať, je potrebné nainštalovať systémové súčasti, ktoré sú využívané k obsluhe použitých funkcií. Nižšie sa nachádza kompletný zoznam príkazov, ktoré tieto súčasti do systému nainštalujú:

```
sudo apt-get update
sudo apt-get install cmake
sudo apt-get install build-essential
sudo apt-get install flex bison
sudo apt-get install libx11-dev libasound-dev libpcap-dev
```

Po nainštalovaní potrebných súčastí je nutné nainštalovať knižnicu **PortAudio**. Pre inštaláciu je nutné v priečinku knižnice otvoriť terminál a zadať príkazy:

```
./configure && make
sudo make install
```

Následne je potrebné povoliť v systéme zachytávanie komunikácie na USB portoch príkazom:

```
sudo modprobe usbmon
```

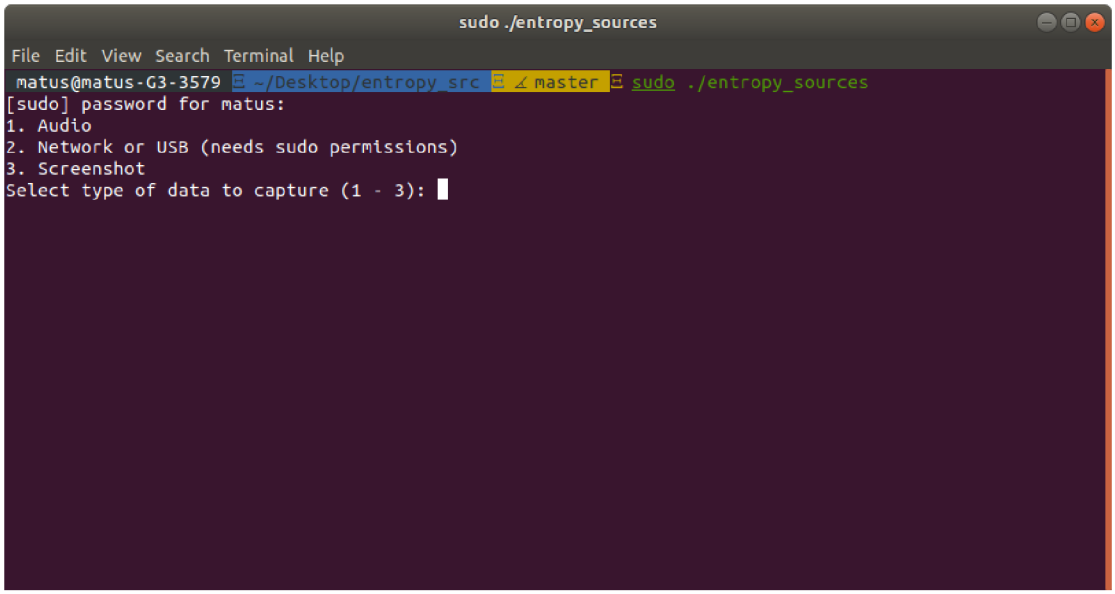
V tomto momente je systém pripravený na kompiláciu samotného programu. V priečinku programu je potrebné zadať príkazy:

```
cmake .
make
```

ktorých výsledkom je spustiteľná verzia programu. Program je potrebné pustiť s administrátorskými právami, čo je možné vykonať príkazom:

```
sudo ./entropy_sources
```

V prípade že sú všetky kroky vykonané v danom poradí, spustí sa program, ktorého úvodnú ponuku je možné vidieť na obrázku 3.1.

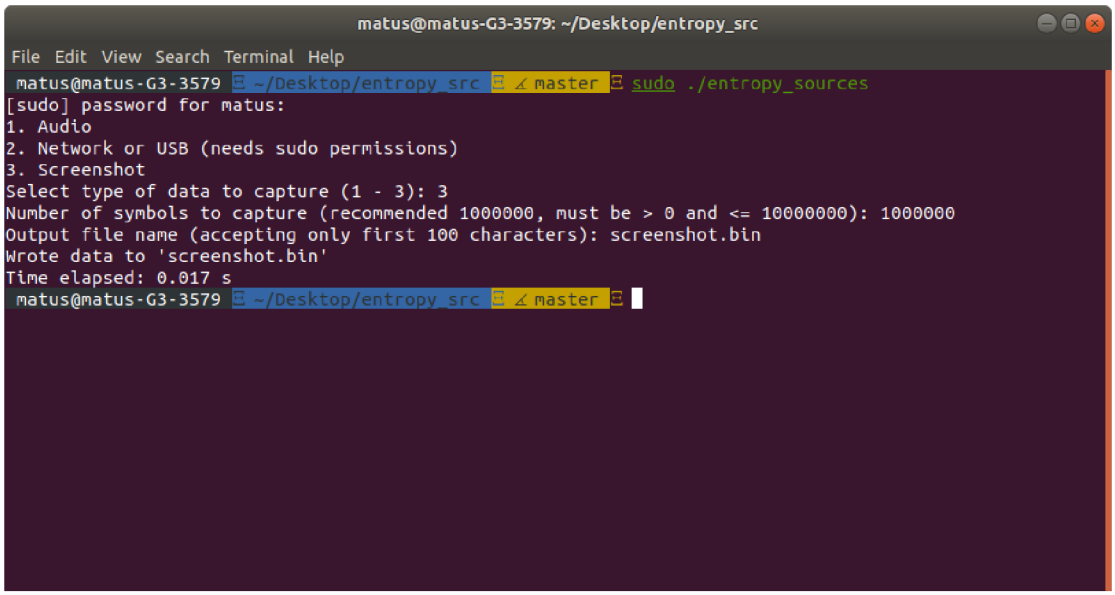


Obr. 3.1: Úvodná obrazovka programu pre získavanie bitových postupností zo zdrojov entropie.

Z ponuky je možné vybrať tri zdroje entropie:

1. zvukovú kartu,
2. sieťovú kartu a USB,
3. obrazový výstup.

Ako je možné vidieť na obrázku 3.2, po výbere zdroja entropie je nutné zadať požadovaný počet znakov a názov súboru, do ktorého budú uložené. Následne je užívateľ informovaný o výsledku jeho požiadavky a o čase trvania záznamu. V prípade,



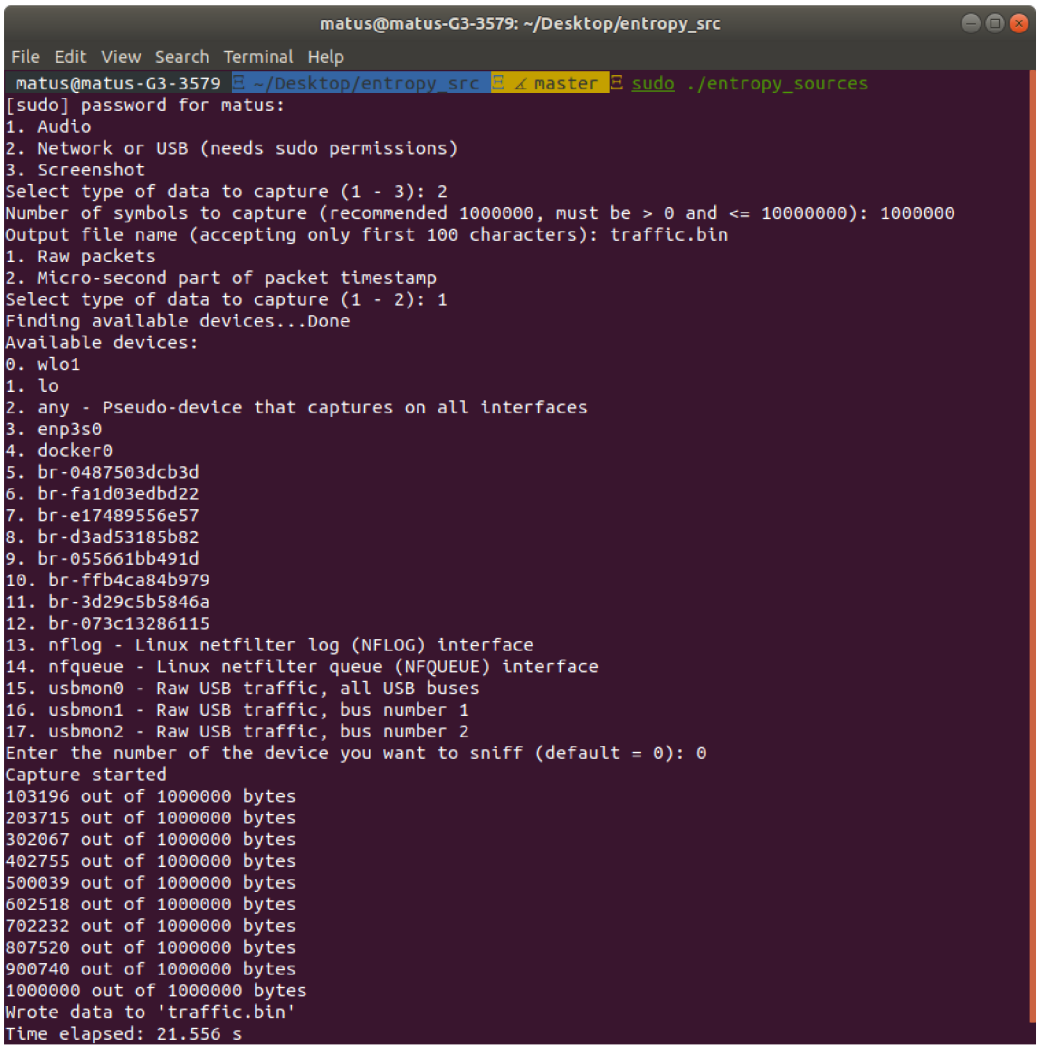
Obr. 3.2: Priebeh procesu získavania bitov zo zdroja entropie.

že je ako zdroj entropie zvolená sieťová karta a USB, je potrebné okrem spomenutých údajov zadať aj typ zbieraných dát (surové dáta (**1**) alebo mikrosekundová časť časového razítka (**2**)). V poslednom kroku je užívateľovi zobrazená ponuka sieťových rozhraní kombinovaná s rozhraniami USB portov. Po výbere rozhrania je zahájené zachytávanie dát, o ktorom je užívateľ pravidelne informovaný. Na obrázku 3.3 je možné vidieť celý priebeh zaznamenávania dát zo sieťového rozhrania.

3.2 Program pre hodnotenie zdrojov entropie

Ako bolo popísané v kapitole 2, hodnotenie zdrojov entropie zahŕňa množstvo testov pre overenie predpokladu nezávislosti a identickej distribúcie dát a ďalšie výpočty slúžiace k odhadu minimálnej entropie, preto bolo jednou z hlavných požiadaviek možnosť paralelného spracovania výpočtov. Medzi ďalšie požiadavky patrila virtualizácia (resp. kontajnerizácia) finálneho riešenia, ktorá umožní spustenie programu bez ohľadu na operačný systém a bez nutnosti vytvárania behového prostredia (angl. *runtime environment*).

Spomedzi množstva programovacích jazykov bol zvolený jazyk **Golang** [34] vo verzii **1.14**, ktorého najväčšou výhodou – s ohľadom na kladené požiadavky – je bohatá podpora súbežnosti (angl. *concurrency*). Súbežnosť umožňuje rozdeliť sekvenčný program na menšie nezávislé časti, ktoré môžu bežať a vykonávať výpočty súčasne s ďalšími časťami, čo sa citeľne odrazí na dobe behu programu. Tým sa zároveň zvýši efektivita využitia procesoru programom.



Obr. 3.3: Priebeh procesu získavania bitov zo sieťovej prevádzky.

3.2.1 Popis programu

Celý program je riadený zo súboru `main.go`, ktorý zabezpečuje vytvorenie mikroservisov pre testovanie IID predpokladu a pre odhad výstupnej entropie. V súbore sa takisto nachádza riadiaca logika programu.

V priečinku `internal` je možné nájsť tri ďalšie podpriečinky:

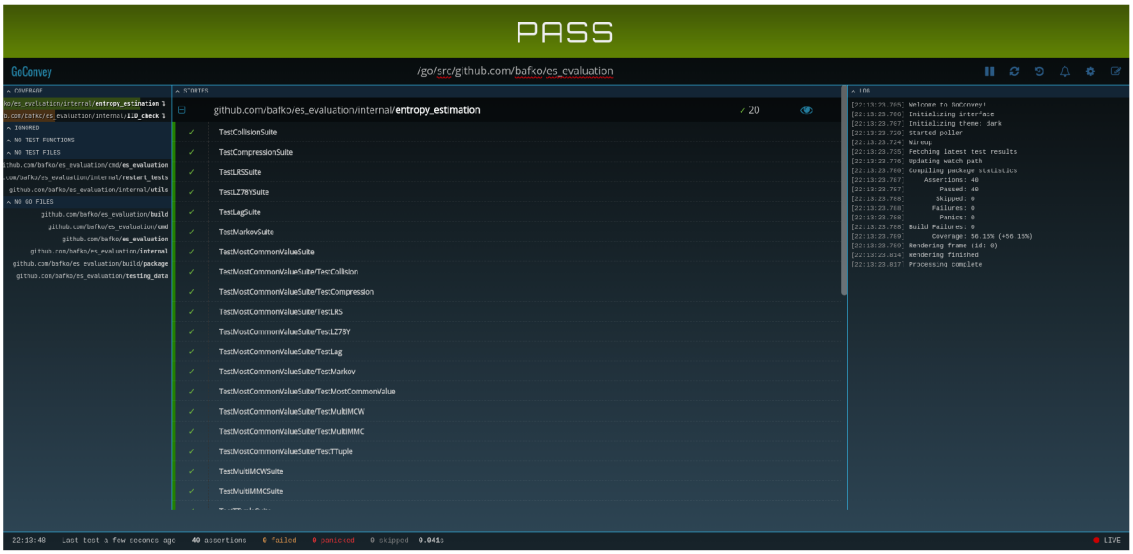
- `entropy_estimation` obsahujúci mikroservis pre odhad výstupnej entropie,
- `IID_check` obsahujúci mikroservis pre testovanie IID predpokladu (permutačné a chi-kvadrát testy),
- `utils` obsahujúci pomocné matematické funkcie (napr. na výpočet chi-kvadrát hodnoty) a funkcie na manipuláciu so súbormi.

Na začiatku programu je načítaný vstupný súbor, zistená veľkosť abecedy a je vy-

konané premapovanie jednotlivých znakov. Následne je spustené overenie IID predpokladu pomocou chi-kvadrát testov. V prípade, že testy vrátia pozitívny výsledok, je IID predpoklad overený ešte permutačnými testami. V prípade, že tento predpoklad nie je vyvrátený ani po permutačných testoch, sú dáta prehlásené za IID a odhad výstupnej entropie je získaný iba odhadom pomocou najčastejšej hodnoty (2.4.1). V prípade že je IID predpoklad vyvrátený, odhad sa riadi postupom popísaným v 2.4.

3.2.2 Testy

V štandarde organizácie NIST SP 800-90B [6] je ku každému odhadu výstupnej entropie krátky príklad, ktorý demonštruje priebeh testu a slúži k overeniu funkčnosti implementácie. Tieto testy sú súčasťou programu a je možné ich spustiť podľa návodu v časti 3.2.4. K testom je dostupné aj webové rozhranie zobrazujúce výsledky testov, ktorého náhľad je možné vidieť na obrázku 3.4.



Obr. 3.4: Výsledok testov funkčnosti popísaných v dokumente SP 800-90B [6].

3.2.3 Kontajnerizácia

Kontajnerizácia programu je riešená prostredníctvom nástrojov **Docker** a **Docker Compose** [34]. Základom je súbor **Dockerfile** (výpis 3.2), ktorý zabezpečí zostavenie výsledného softvérového obrazu, ktorý je následne spustený v samostatnom kontajneri. V prvom kroku sa nastaví pracovný priečinok. Následne sa prekopírujú súbory projektu, stiahnu sa závislosti a program sa zostaví. Výsledný spustiteľný súbor

je následne presunutý do prázdneho obrazu, ktorý je zároveň výsledným obrazom.

Výpis 3.2: Obsah súboru `Dockerfile`.

```
# build workspace
FROM golang:1.14 as build

WORKDIR /go/src/github.com/bafko/es_evaluation

COPY . .

RUN go mod vendor

# For scratch prod builds
RUN CGO_ENABLED=0 GOOS=linux go install -a ./cmd/es_evaluation

# prod build
FROM scratch

WORKDIR /go/bin
ENV PATH=/bin

COPY --from=build /go/bin/es_evaluation .
```

Výsledný softvérový obraz je spustený programom **Docker Compose**, ktorého definície sa nachádzajú v súbore `docker-compose.yaml` (výpis 3.3). V súbore je definovaný obraz na zostavenie, k obrazu je pripojený priečinok so súbormi na testovanie a je nastavená systémová premenná, v ktorej je zadávaný názov súboru určeného k testovaniu.

Výpis 3.3: Obsah súboru `docker-compose.yaml`.

```
version: '3'
services:
  es_evaluation:
    restart: on-failure
    build:
      context: .
      dockerfile: ./build/package/Dockerfile
    volumes:
      - ./testing_data:/go/bin/testing_data
    entrypoint: ./es_evaluation
    environment:
      - FILE=$FILE
```

3.2.4 Spustenie programu

Spúšťanie všetkých častí programu je riešené prostredníctvom súboru `Makefile`. K spusteniu programu je potrebné mať nainštalovaný **Docker** a **Docker Compose** a zadať príkaz:

```
make FILE=testing_data/xxx
```

kde `xxx` reprezentuje názov súboru k otestovaniu. K projektu je v priečinku `testing_data` priložených 5 testovacích súborov:

- `data_1.bin`,
- `data_2.bin`,
- `data_3.bin`,
- `data_4.bin`,
- `data_5.bin`.

Výsledný príkaz potom vyzerá nasledovne:

```
make FILE=testing_data/data_5.bin
```

V prípade že obraz programu nie je vytvorený, po prvom spustení je vykonané jeho zostavenie, ktorého priebeh je možné vidieť na obrázku 3.5. Priebeh samotného testovania je možné vidieť na obrázkoch 3.6 a 3.7. Užívateľ je o priebehu a výsledku odhadov informovaný.

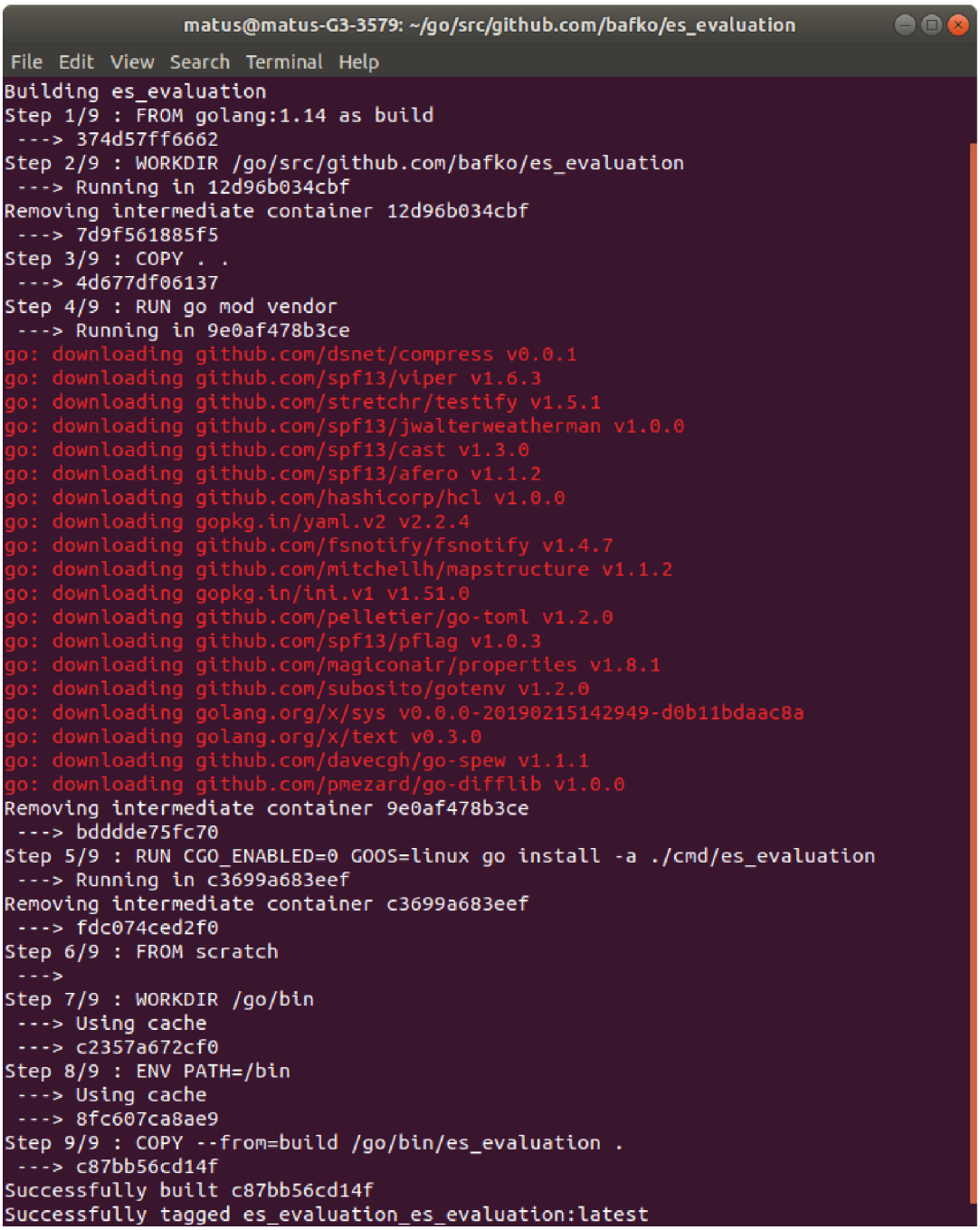
Pre spustenie testov je potrebné mať nainštalovaný **Golang** vo verzii **1.14** alebo vyššej a pred pustením zadať príkaz:

```
go mod vendor
```

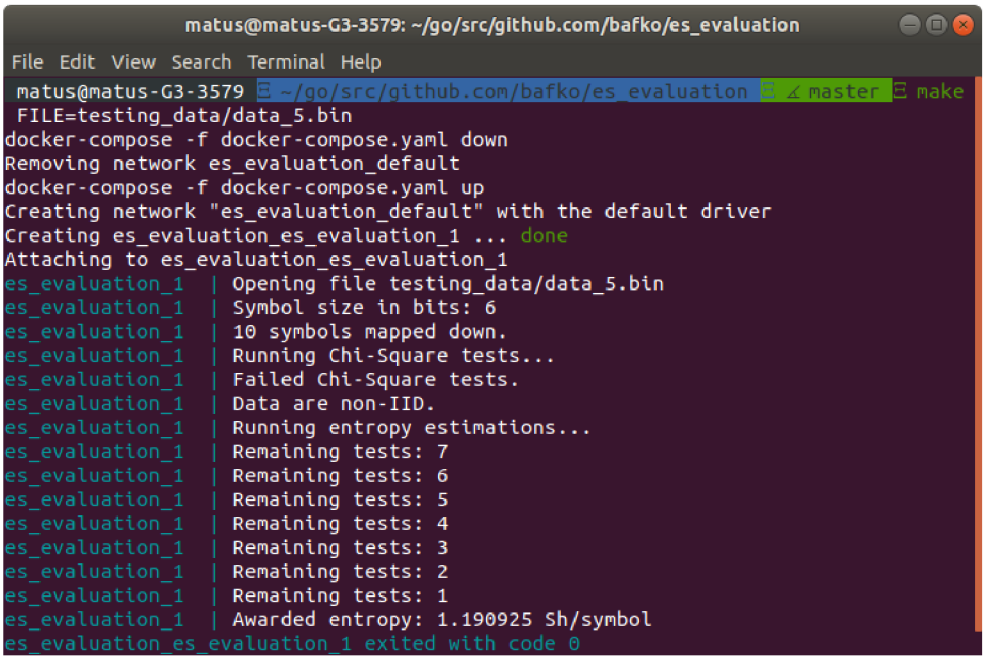
pre stiahnutie závislostí. Spustenie testov je potom možné príkazom:

```
make test
```

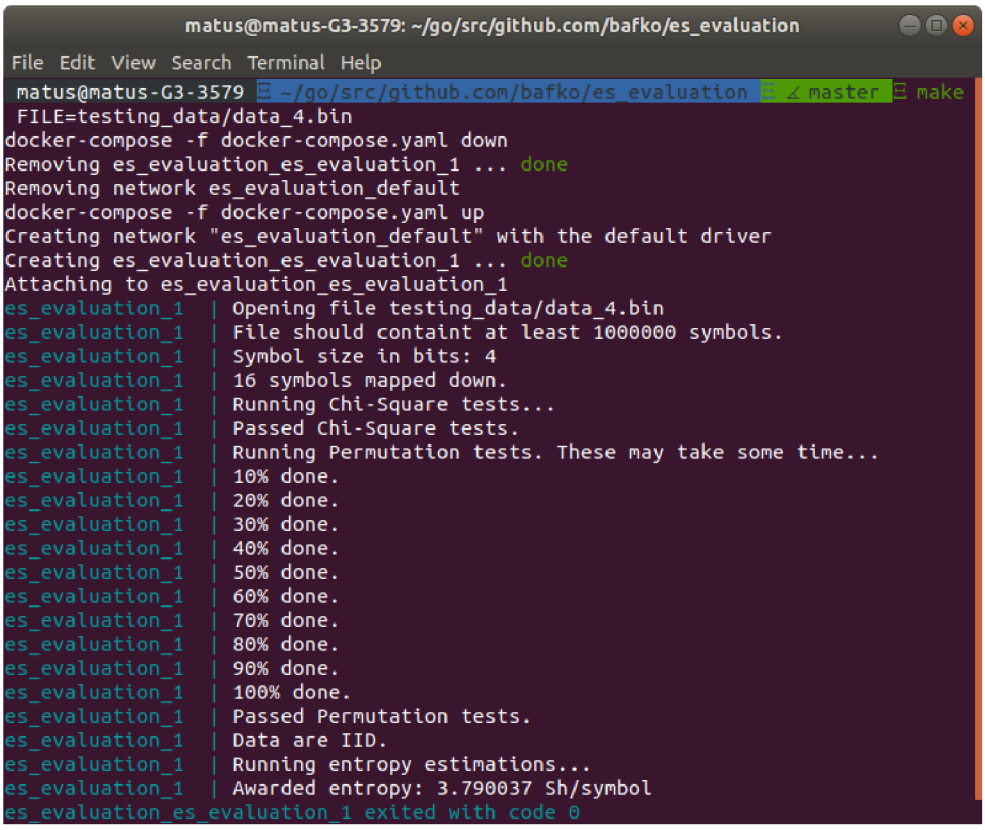
Webové rozhranie testov je možné nájsť na adrese `http://localhost:8080`.



Obr. 3.5: Zostavenie obrazu obsahujúceho spustiteľný program.



Obr. 3.6: Priebeh hodnotenia zdroja entropie.



Obr. 3.7: Priebeh hodnotenia zdroja entropie pri IID dátach.

4 Výsledky hodnotenia zdrojov entropie

Táto kapitola bude zameraná na porovnanie zdrojov entropie, z ktorých je prostredníctvom vytvoreného programu možné zbierať dáta. **Všetky údaje v tabuľkách sú výsledkom aritmetického priemeru troch nezávislých realizácií získavania bitov. Každý zo súborov obsahuje 1 000 000 znakov z daného zdroja entropie.** Jeden znak môže byť veľkosti 1 až 8 bitov v závislosti od zdroja.

4.1 Vstupné dáta

V tabuľke 4.1 sa nachádza základný prehľad parametrov jednotlivých zdrojov entropie. Prvých dvanásť riadkov reprezentuje dáta získané zo zvukovej karty v troch rôznych prostrediach (tichá miestnosť – *silence*, ruch v pozadí – *background*, hluk v miestnosti – *noise*) uložených v 4 rôznych dátových typoch (`float32`, `int32`, `int16` a `uint8`). Dátový formát `int8` bol z tabuľky zámerne vynechaný, nakoľko v ďalších meraniach vykazoval nulovú entropiu.

Riadok *screenshot* reprezentuje dáta získane zo snímky aktuálneho obsahu obrazovky.

Riadky začínajúce kľúčovým slovom *network* reprezentujú dáta získané zo sieťovej karty. Skúmané boli dáta z troch typov prevádzky:

1. *idle* – bez aktivity užívateľa,
2. *browsing* – prehliadanie webových stránok,
3. *downloading* – sťahovanie súboru (približnou rýchlosťou 130 Mbit/s).

Posledná časť názvu súboru reprezentuje typ zbieraných dát, *data* znamenajú surové dáta (resp. pakety), *time* znamená zaznamenávanie mikrosekundovej časti časového razítka prichádzajúceho paketu. Kombinácia *idle* a *time*, bola z tabuľky vynechaná, získavanie bitov sa ani po niekoľkých hodinách nepodarilo dokončiť.

Podobnými pravidlami sa riadi aj zaznamenávanie dát z USB portov, kde bolo takisto možné zaznamenávať surové dáta (*data*) alebo mikrosekundovú časť časového razítka (*time*). Počas merania boli v USB portoch zapojené klávesnica a myš.

Pre porovnanie boli získané aj dáta z hybridného zdroja entropie – linuxového generátoru pseudonáhodných čísel **`urandom`** – prostredníctvom príkazu:

```
head -c 1000000 </dev/urandom > urandom.bin
```

Ako bolo spomenuté, jedná sa o generátor pseudonáhodných čísel, ktorý je inicializovaný semenom z linuxového zdroja entropie. Tento zdroj zbiera entropiu z:

- klávesnice,
- myši,

- systémových prerušení,
- disku,

a ukladá ju do zásobníku o veľkosti 4 096 bitov. Z tohto zásobníku následne čerpajú generátory **random** (ktorý je priamo závislý na množstve nepredvídateľných bitov v zásobníku a v prípade ich nedostatku je blokujúci) a **urandom** (resp. funkcia `get_random_bytes`), ktorý entropiu zo zásobníku využíva ako inicializačné semeno v generátore (konkrétne dve 32-bitové slová). V prípade, že je v zásobníku nedostatok nepredvídateľných bitov je `urandom` neblokujúci (pozn.: po štarte systému je v rovnakom stave blokujúci), čo v extrémnych prípadoch môže mať za následok opakovanie výstupných sekvencií bitov. V systéme je bežne používaný ako zdroj nepredvídateľných bitov pre kryptografické účely, preto je ako názorná ukážka zaradený medzi zdroje entropie [36]. Nakoľko sa nejedná o základný zdroj entropie, hodnoty v nasledujúcich tabuľkách budú v porovnaní s ostatnými zdrojmi výrazne vyššie.

Prvý stĺpec tabuľky je určený pre názov súboru, z ktorého je možné určiť typ dát. V druhom stĺpci sa nachádza doba zaznamenávania 1 000 000 znakov v sekundách. Tretí stĺpec indikuje počet bitov v jednom znaku. Štvrtý a piaty stĺpec sú určené údajom o rýchlosti zdroja, ktoré vznikli ako podiel počtu zaznamenaných znakov a doby zaznamenávania, resp. podiel podiel počtu zaznamenaných bitov a doby zaznamenávania.

4.2 Prvotné hodnotenie dát

Zdroje boli podrobené testom uvedeným v štandarde organizácie NIST SP 800-90B [6] (aj v tomto prípade sú hodnoty aritmetickým priemerom 3 realizácií). Merania boli vykonané dvoma spôsobmi, prvý z nich hodnotil znaky tak ako sa objavili na výstupe zdroja entropie, druhý spôsob zahŕňal prevod znakov na bity (resp. binárne znaky) a následné hodnotenie dát po tomto prevode. Výsledné hodnoty je možné nájsť v tabuľke 4.2.

V tabuľke 4.2 je rozdelenie zdrojov entropie až príliš jemné a nereprezentuje všeobecné správanie daného zdroja, preto je nutné počet hodnotených zdrojov zredukovať. V tabuľke 4.3 je možné vidieť množstvo entropie získaného zo zvukovej karty v závislosti od okolia a použitého dátového typu pre uloženie dát.

Na obrázku 4.1 je možné vidieť hodnoty z tabuľky 4.3 vynesené do grafu. Z daného grafu je možné vidieť, že pri všetkých formátoch okrem `uint8` bolo množstvo entropie najvyššie v prípade, kedy bol v miestnosti ruch v pozadí. Takisto je z grafu možné vidieť, že výstupné znaky mali najväčšie množstvo entropie pri použití dátového typu `int16`. V nasledujúcich porovnaniach a tabuľkách budú údaje reprezentujúce zvukovú kartu ako zdroj entropie reprezentovať dáta vo formáte `int16`.

Tab. 4.1: Základné informácie o spracúvaných súboroch o veľkosti 1 000 000 znakov.

Názov súboru	Trvanie [s]	Bitov na znak	Rýchlosť [znak/s]	Rýchlosť [bit/s]
audio_silence_float32	3,005	8	332 742	2 661 934
audio_silence_int32	3,004	8	332 853	2 662 820
audio_silence_int16	6,008	8	166 445	1 331 558
audio_silence_uint8	12,007	8	83 285	666 278
audio_background_float32	3,005	8	332 779	2 662 230
audio_background_int32	3,007	8	332 594	2 660 754
audio_background_int16	6,006	8	166 500	1 332 001
audio_background_uint8	12,004	8	83 306	666 445
audio_noise_float32	3,005	8	332 816	2 662 525
audio_noise_int32	3,005	8	332 816	2 662 525
audio_noise_int16	6,007	8	166 482	1 331 854
audio_noise_uint8	12,007	8	83 282	666 260
screenshot	0,019	8	52 631 579	421 052 632
network_idle_data	1 350,279	8	741	5 925
network_browsing_data	6,831	8	146 384	1 171 074
network_downloading_data	0,060	8	16 574 586	132 596 685
network_browsing_time	931,393	6	1 074	6 442
network_downloading_time	10,048	6	99 526	597 154
usb_data	380,540	8	2 628	21 023
usb_time	4 744,856	6	211	1 265
`urandom`	0,010	8	100 000 000	800 000 000

Hodnoty entropie a parametre zdroja budú aritmetickým priemerom všetkých troch prostredí, čo by malo pokryť možné scenáre prebiehajúce v okolí zvukového vstupu (mikrofónu).

Rovnakým spôsobom akým boli porovnané dáta zo zvukovej karty sú v tabuľke 4.4 porovnané údaje dát získaných zo sieťovej prevádzky. Z grafov vynesených v obrázku 4.2 je možné vidieť, že s množstvom sieťovej prevádzky rástlo aj množstvo entropie na výstupe zdroja. Tento stav však nie je možné zaistiť, preto bude v nasledujúcich porovnaniach uvažovaný aritmetický priemer z hodnôt všetkých troch typov dátovej prevádzky. Pri sieťovej a USB prevádzke zostane v porovnaní aj *time*, aj *data*, nakoľko je pri zaznamenávaní dát riziko útoku lokálnym útočníkom. Zaznamenávanie mikrosekundovej časti časového razítka času príchodu paketu túto nevý-

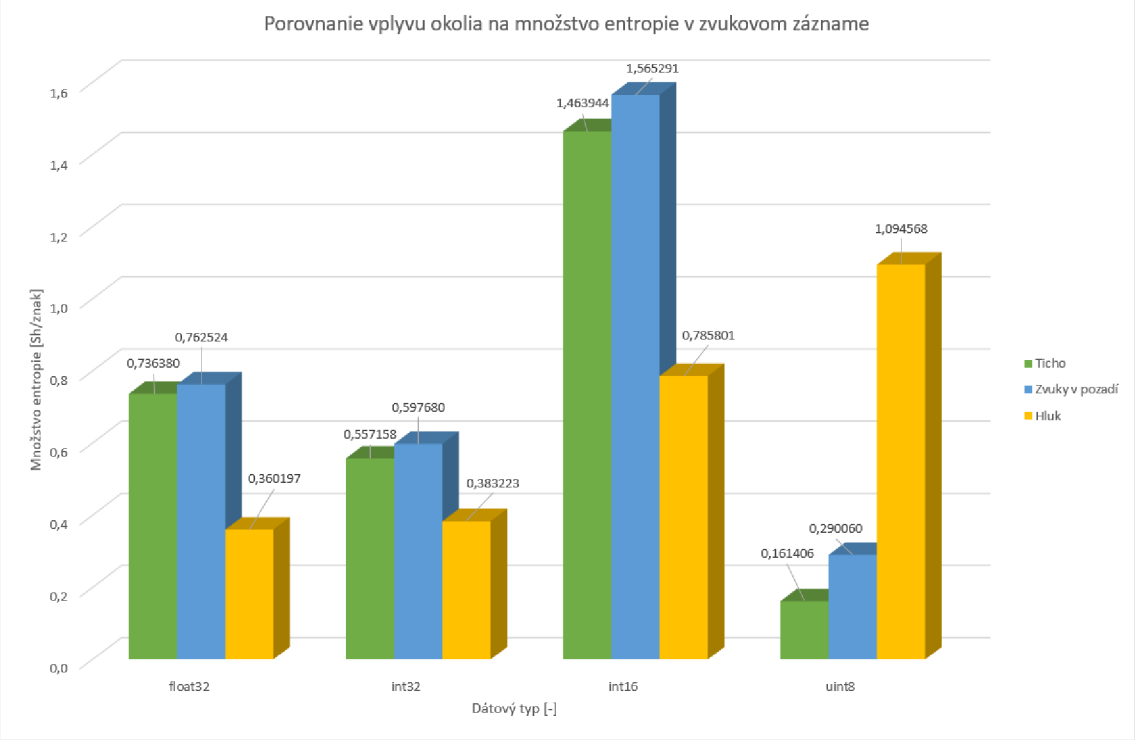
Tab. 4.2: Entropia testovaných súborov.

Názov súboru	Entropia [Sh/znak]	Entropia [Sh/bit]
audio_silence_float32	0,736380	0,169489
audio_silence_int32	0,557158	0,113108
audio_silence_int16	1,463944	0,251747
audio_silence_uint8	0,161406	0,139640
audio_background_float32	0,762524	0,174816
audio_background_int32	0,597680	0,122257
audio_background_int16	1,565291	0,278434
audio_background_uint8	0,290060	0,041932
audio_noise_float32	0,360197	0,054457
audio_noise_int32	0,383223	0,058067
audio_noise_int16	0,785801	0,120109
audio_noise_uint8	1,094568	0,131167
screenshot	0,001159	0,000200
network_idle_data	0,016886	0,002390
network_browsing_data	0,021462	0,003049
network_downloading_data	0,597298	0,092253
network_browsing_time	0,759847	0,095374
network_downloading_time	0,671121	0,094452
usb_data	0,136815	0,024153
usb_time	1,187794	0,099118
`urandom`	7,880298	0,998237

Tab. 4.3: Porovnanie množstva entropie v zvukových nahrávkach.

Dátový typ → Popis prostredia ↓	float32 [Sh/znak]	int32 [Sh/znak]	int16 [Sh/znak]	uint8 [Sh/znak]
Ticho	0,736380	0,557158	1,463944	0,161406
Ruch v pozadí	0,762524	0,597680	1,565291	0,290060
Hluk	0,360197	0,383223	0,785801	1,094568

hodu odstraňuje, nakoľko čas príchodu a spracovania paketu je závislý od množstva okolností a pre užívateľa (resp. útočníka) nepredvídateľný.



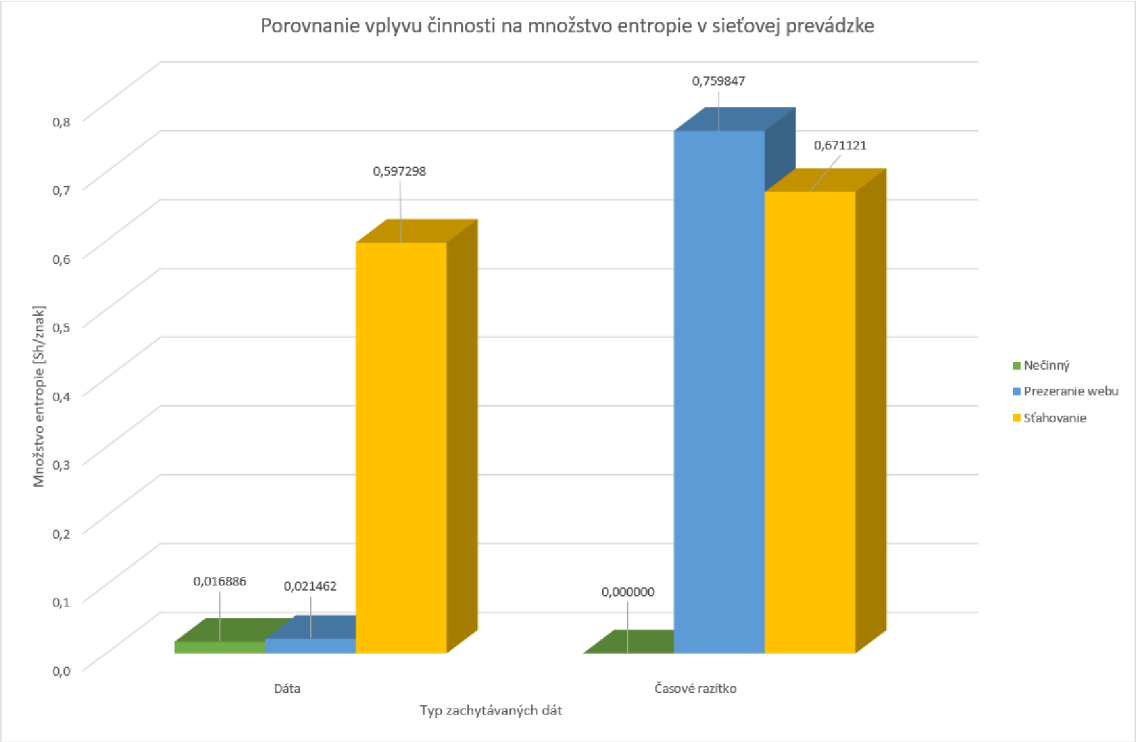
Obr. 4.1: Porovnanie vplyvu okolia na množstvo entropie v zvukovom zázname.

Tab. 4.4: Porovnanie množstva entropie z rôznych druhov sieťovej prevádzky.

Typ zbieraných dát → Popis činnosti ↓	Surové dáta [Sh/znak]	Časové razítko [Sh/znak]
Nečinný	0,016886	—
Prezeranie webu	0,021462	0,759847
Sťahovanie súborov	0,597298	0,671121

4.3 Porovnanie zdrojov entropie

Po zredukovaní množstva porovnávaných údajov zostalo 7 zdrojov entropie, ktorých základné údaje, vrátane množstva entropie na výstupe, je možné vidieť v tabuľke 4.5. Pokiaľ nezoberieme do úvahy linuxový generátor pseudonáhodných čísel `urandom`, zdrojom s najväčším množstvom entropie na výstupe je zvuková karta, za ňou nasleduje prevádzka na USB portoch (pri zaznamenávaní mikrosekundovej časti časového razítka) a sieťová prevádzka (takisto pri zaznamenávaní mikrosekundovej časti časového razítka). S veľkým odstupom nasledujú dáta zo sieťovej prevádzky, dáta z komunikácie na USB portoch a na konci sa umiestnila snímka obrazovky



Obr. 4.2: Porovnanie vplyvu činnosti na množstvo entropie v sieťovej prevádzke.

s entropiou o dva rády nižšou oproti predošlému zdroju.

Tab. 4.5: Porovnanie vybraných zdrojov entropie pri vzorke 1 000 000 znakov.

Názov zdroja	Trvanie [s]	Rýchlosť [znak/s]	Rýchlosť [bit/s]	Entropia [Sh/znak]	Entropia [Sh/bit]
usb_time	4 744,856	211	1 265	1,187794	0,099118
usb_data	380,540	2 628	21 023	0,136815	0,024153
network_data	452,390	2 210	17 684	0,211882	0,032564
network_time	470,721	2 124	12 746	0,715484	0,094913
screenshot	0,019	52 631 579	421 052 632	0,001159	0,000200
audio	6,007	166 476	1 331 804	1,271679	0,216763
`urandom`	0,010	100 000 000	800 000 000	7,880298	0,998237

Dôležité je však zobrať do úvahy aj rýchlosť generovania bitov, ktorá dokáže vykompenzovať nedostatok entropie na výstupe. Pre tento účel nám poslúži veličina **výdatnosť**, ktorá je definovaná ako súčin entropie zdroja a jeho rýchlosti. Táto veličina zohľadňuje všetky merateľné parametre zdroja entropie a bude považovaná za

indikátor kvality zdroja, na základe ktorého bude vykonané ich hodnotenie. V tabuľke 4.6 je možné vidieť výdatnosť jednotlivých zdrojov entropie – pre dáta spracúvané ako znaky, ale aj ako bitovú postupnosť.

V tabuľke sú zdroje entropie zoradené vzostupne podľa ich výdatnosti. Na poslednom mieste sa umiestnila prevádzka na USB portoch. Verzia zaznamenávania času síce vykazovala vyššiu entropiu, avšak vďaka vyššej rýchlosti generovania výstupných znakov sa varianta *data* umiestnila na vyššej priečke.

Nasleduje sieťová prevádzka, ktorej varianta *time* dokáže generovať rádovo tisícky Shannonov za sekundu (ak veličinu Shannon prevedieme do intuitívnejšej podoby, pod pojmom jeden Shannon si je s istými zanedbaniami možné predstaviť jeden **skutočne náhodný** bit alebo znak).

Je dôležité poznamenať, že hodnoty získané z USB a sieťovej prevádzky sú individuálne a výrazne závislé od aktivity systému a užívateľa. Súčasne je zdroj *network_time* značne nadhodnotený, nakoľko dáta pri nečinnosti neboli získané. Cieľom tohto porovnania nie je exaktné vyčíslenie výdatnosti, ale porovnanie zdrojov z hľadiska dlhodobej prevádzky, čo získané dáta dostatočne reprezentujú.

Na predposlednom mieste (pri zanedbaní `urandom`) sa umiestnila snímka obrazovky. Posledné miesto v porovnaní entropie je v tomto prípade vykompenzované vysokou rýchlosťou získavania dát zo zdroja entropie.

Na prvom mieste sa umiestnila zvuková karta, ktorá predošlý zdroj entropie prekonala viac ako trojnásobne a je schopná generovať stovky tisíc Shannonov informácie za sekundu. Zároveň sa jedná o zdroj, ktorého správanie a rýchlosť je z dlhodobého hľadiska stabilná a veľmi ťažko ovplyvniteľná útočníkom. **Zvuková karta** preto bude v rámci práce označená za **najlepší zdroj entropie**, ktorý bol skúmaný.

Tab. 4.6: Porovnanie výdatnosti vybraných zdrojov entropie pri vzorke 1 000 000 znakov.

Názov zdroja	Výdatnosť - znaky [Sh/s]	Výdatnosť - bity [Sh/s]
usb_time	250	125
usb_data	360	508
network_data	468	576
network_time	1 520	1 210
screenshot	61 000	84 168
audio	211 703	288 686
`urandom`	788 029 767	798 589 600

4.4 Vplyv hešovania na množstvo entropie

Poslednou skúmanou časťou boli metódy zvýšenia množstva entropie na výstupe a tým aj výdatnosti samotného zdroja. Ako efektívny nástroj na zvýšenie množstva entropie boli zvolené hešovacie funkcie a to konkrétne **MD5** (spracovanie/skrátenie správy – *Message Digest*), **SHA-1** (bezpečný hešovací algoritmus – *Secure Hash Algorithm*), **SHA-256** a **SHA-512**. Metodika merania zvýšenia množstva entropie bola rovnaká ako pri hodnotení zdrojov entropie – hodnoty v tabuľkách sú aritmetickým priemerom troch nezávislých realizácií. Hešovacie funkcie boli aplikované vždy na počet znakov, ktorý sa rovná počtu znakov na jej výstupe (napríklad pri MD5, ktorá produkuje na výstupe 16 znakov bolo k hešovaniu použitých 16 znakov zo zdroja entropie). Následne boli ohodnotené dáta vzniknuté týmto postupom – v podobe znakov, ale aj po prevedení na binárne symboly. Výslednú entropiu na výstupe po aplikovaní jednotlivých hešovacích funkcií je možné nájsť v tabuľkách 4.7 a 4.8.

Po prepočítaní relatívneho zvýšenia entropie (vzhľadom k pôvodnej entropii) a vypočítaní aritmetického priemeru jednotlivých zvýšení v rámci danej hešovacej funkcie dostávame prehľadné porovnanie „kvality“ hešovacích funkcií, ktoré je možné vidieť v prehľadnom grafe na obrázku 4.3. Najväčšie relatívne zvýšenie množstva entropie dosiahla hešovacia funkcia SHA-512. Za ňou nasledujú SHA-256, SHA-1 a MD5. Z grafu je takisto možné vidieť, že entropia sa zvýšila viac, pokiaľ sa výsledný súbor hodnotil podľa znakov a nie podľa jednotlivých bitov. Nakoľko funkcia SHA-512 výrazne prekonala ostatné hešovacie funkcie, bude voči pôvodným dátam porovnaná práve ona.

V tabuľke 4.9 je možné vidieť výsledné entropie a výdatnosti zdrojov po aplikovaní funkcie SHA-512. Následne je v tabuľke 4.10 možné vidieť porovnanie zdrojov entropie na základe výdatnosti pred a po aplikovaní hešovacej funkcie SHA-512, a to pri hodnotení dát ako znakov alebo bitov. Za zmienku stojí, že aplikovanie funkcie SHA-512 na zdroj `urandom` malo zanedbateľný vplyv na množstvo výstupnej entropie, čo svedčí o kvalite samotného zdroja (resp. generátoru). Na záver je možné na obrázkoch 4.4 (hodnotenie dát ako znakov) a 4.5 (hodnotenie dát ako bitov) možné vidieť vplyv hešovacej funkcie na výslednú výdatnosť zdrojov. **Grafy majú logaritmickú mierku**.

Nakoľko do hešovacích funkcií nevstupuje žiadna dodatočná entropia a nebola vykonaná ani **koncentrácia entropie** (napríklad 160 znakov s entropiou 0,1 Sh/znak skoncentrovaných pomocou hešovacej funkcie do výstupných 16 znakov s entropiou 1 Sh/znak), je možné prehlásiť, že hešovacia funkcia ovplyvnila iba samotné rozloženie dát, čo malo za následok odlišný priebeh odhadov a tým aj odlišnú odhadovanú entropiu. Tento záver je možné podložiť aj postupom, kedy by sme na vstupe mali

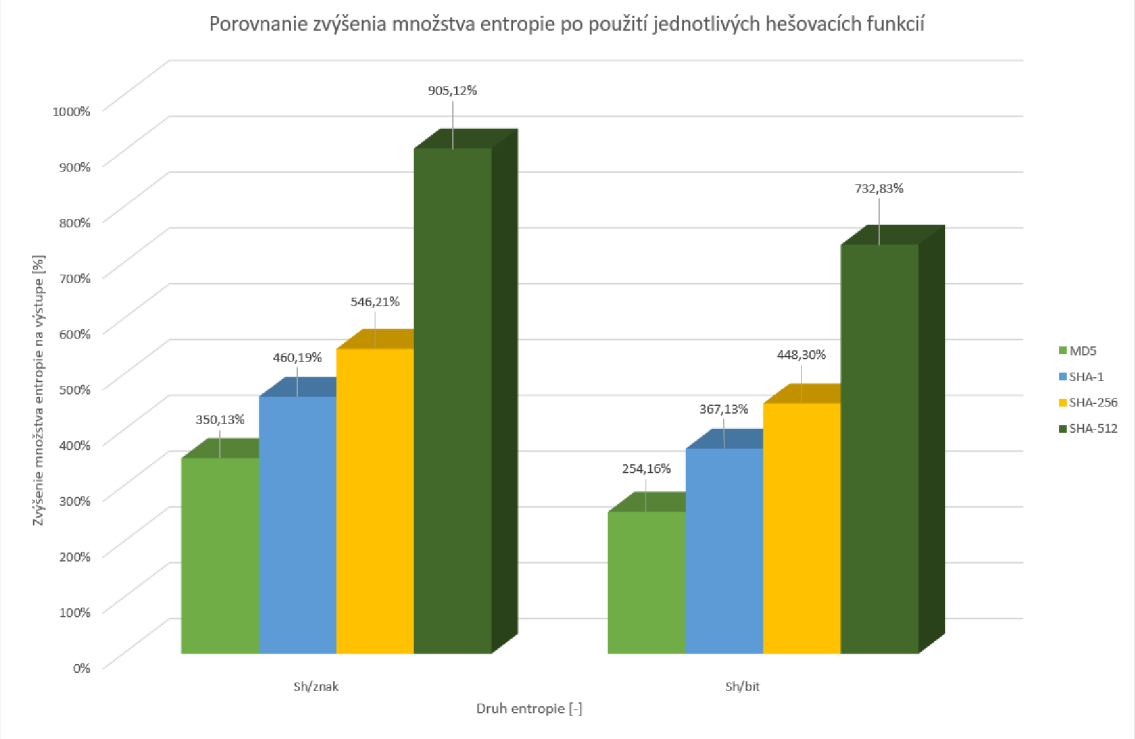
Tab. 4.7: Entropia testovaných súborov po aplikovaní hešovacej funkcie MD5 a SHA-1.

	MD5		SHA-1	
Názov súboru	Entropia [Sh/znak]	Entropia [Sh/bit]	Entropia [Sh/znak]	Entropia [Sh/bit]
audio_silence_float32	7,464987	0,973332	7,874201	0,998093
audio_silence_int32	7,879617	0,998348	7,885274	0,998433
audio_silence_int16	7,874539	0,998458	7,881489	0,998066
audio_silence_uint8	5,868832	0,750799	5,975392	0,760518
audio_background_float32	7,887363	0,998506	7,871812	0,998359
audio_background_int32	6,065689	0,776857	7,876735	0,998273
audio_background_int16	7,871607	0,970158	7,881920	0,998408
audio_background_uint8	0,540129	0,091869	0,559057	0,094084
audio_noise_float32	0,528483	0,081679	0,640594	0,098822
audio_noise_int32	0,556888	0,087046	0,681611	0,103953
audio_noise_int16	1,571897	0,226172	1,668694	0,242028
audio_noise_uint8	5,985132	0,768235	7,884348	0,998501
screenshot	0,001444	0,000233	0,001489	0,000238
network_idle_data	0,019285	0,002783	0,016461	0,002823
network_browsing_data	0,056413	0,008216	0,069480	0,010949
network_downloading_data	0,880163	0,133475	1,602899	0,225996
network_browsing_time	6,053336	0,778817	7,702766	0,973579
network_downloading_time	2,342119	0,328208	7,869264	0,998428
usb_data	0,233992	0,037044	0,243240	0,037682
usb_time	2,334592	0,329652	7,879507	0,998470
`urandom`	7,875809	0,998379	7,881732	0,998379

napríklad 1,6 Sh informácie (16 8-bitových znakov s entropiou 0,1 Sh/znak) a po aplikovaní hešovacej funkcie 120 Sh informácie (16 8-bitových znakov s entropiou 7,5 Sh/znak) na výstupe. Z pohľadu odhadov by prípadný útočník potreboval informáciu o hodnote 120 Sh (resp. „uhádnuť“ 120 bitov), pričom v skutočnosti by mu na zreprodukovanie výsledkov stačilo iba 1,6 Sh („uhádnuť“ 1,6 bitu). Cieľom tejto časti je poukázať na nedokonalosti v metodike testovania zdrojov entropie založenej na štandarde SP 800-90B [6].

Tab. 4.8: Entropia testovaných súborov po aplikovaní hešovacej funkcie SHA-256 a SHA-512.

	SHA-256		SHA-512	
Názov súboru	Entropia [Sh/znak]	Entropia [Sh/bit]	Entropia [Sh/znak]	Entropia [Sh/bit]
audio_silence_float32	7,882407	0,998460	7,879499	0,998495
audio_silence_int32	7,873738	0,998148	7,695229	0,964080
audio_silence_int16	7,883074	0,998360	7,876153	0,998386
audio_silence_uint8	6,010219	0,763806	6,008561	0,760222
audio_background_float32	7,889339	0,998057	7,874528	0,998097
audio_background_int32	7,873733	0,998132	7,878907	0,998522
audio_background_int16	7,876147	0,998274	7,873967	0,998430
audio_background_uint8	0,922943	0,134373	7,876605	0,998260
audio_noise_float32	0,726990	0,111200	7,881026	0,992121
audio_noise_int32	3,247712	0,420003	7,883209	0,998433
audio_noise_int16	7,881450	0,998422	7,881688	0,998110
audio_noise_uint8	7,704088	0,968351	7,881689	0,998520
screenshot	0,001580	0,000243	0,001713	0,000243
network_idle_data	0,020044	0,002926	0,023114	0,003178
network_browsing_data	0,070936	0,012411	0,109162	0,016203
network_downloading_data	3,413367	03441321	7,879254	0,998192
network_browsing_time	5,543612	0,998427	7,880758	0,998192
network_downloading_time	7,880543	0,998259	7,879141	0,998230
usb_data	0,382309	0,059494	0,552439	0,075901
usb_time	7,886703	0,998445	7,878130	0,998351
`urandom`	7,883881	0,998460	7,874891	0,998519



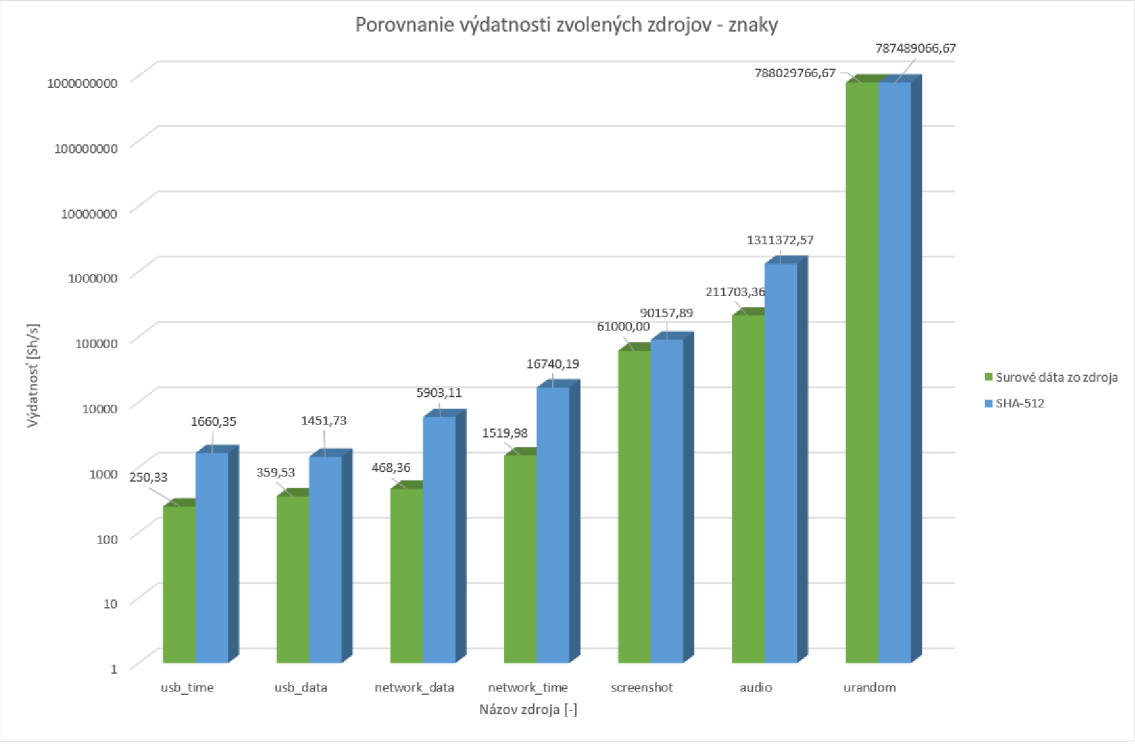
Obr. 4.3: Porovnanie zvýšenia množstva entropie po použití jednotlivých hešovacích funkcií.

Tab. 4.9: Porovnanie entropie a výdatnosti vybraných zdrojov entropie po použití hešovacej funkcie SHA-512.

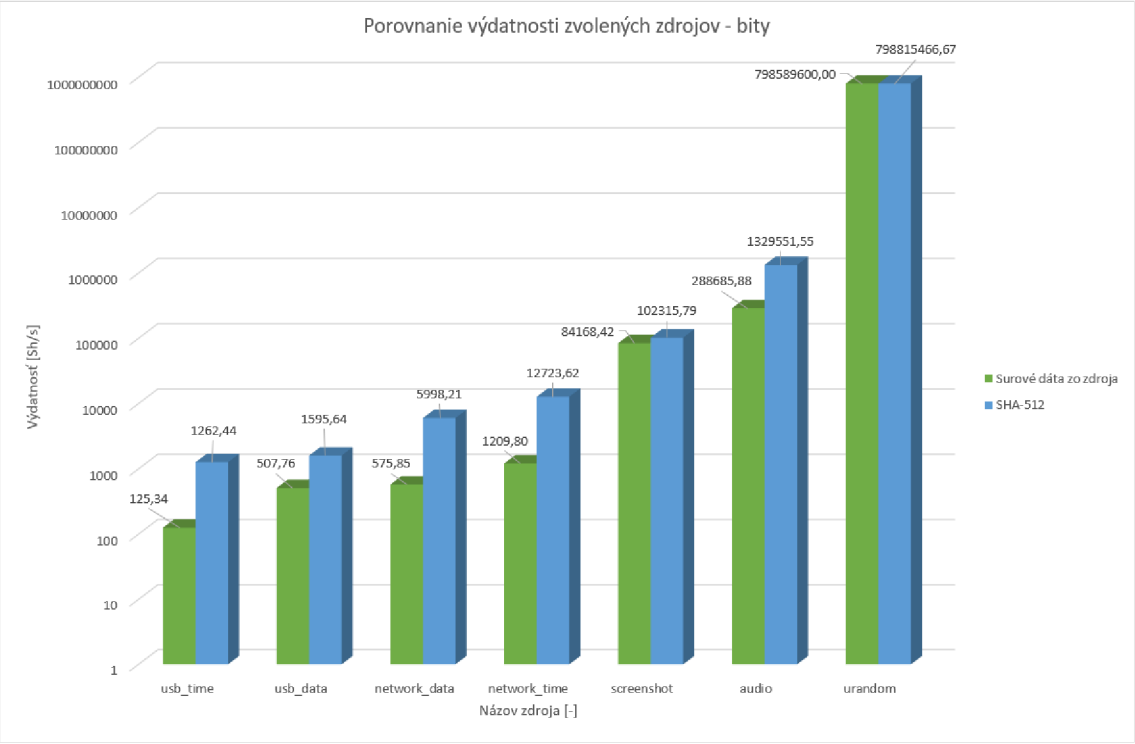
Názov zdroja	Entropia [Sh/znak]	Entropia [Sh/bit]	Výdatnosť - znaky [Sh/s]	Výdatnosť - bity [Sh/s]
usb_time	7,878130	0,998351	1 660	1 262
usb_data	0,552439	0,0759601	1 452	1 596
network_data	2,670510	0,339191	5 903	5 998
network_time	7,879950	0,998211	16 740	12 724
screenshot	0,001713	0,000243	90 158	102 316
audio	7,877269	0,998309	1 311 373	1 329 552
`urandom`	7,874891	0,998519	787 489 067	798 815 467

Tab. 4.10: Záverečné porovnanie výdatnosti vybraných zdrojov entropie pred a po použití hešovacej funkcie SHA-512.

	Pôvodné dáta		SHA-512	
Názov zdroja	Výdatnosť - znaky [Sh/s]	Výdatnosť - bity [Sh/s]	Výdatnosť - znaky [Sh/s]	Výdatnosť - bity [Sh/s]
usb_time	250	125	1 660	1 262
usb_data	360	508	1 452	1 596
network_data	468	576	5 903	5 998
network_time	1 520	1 210	16 740	12 724
screenshot	61 000	84 168	90 158	102 316
audio	211 703	288 686	1 311 373	1 329 552
`urandom`	788 029 767	798 589 600	787 489 067	798 815 467



Obr. 4.4: Porovnanie výdatnosti zvolených zdrojov - znaky.



Obr. 4.5: Porovnanie výdatnosti zvolených zdrojov - bity.

Záver

Zdroje entropie a ich hodnotenie za sebou skrýva bohatú problematiku, ktorú dopodrobna popisuje štandard organizácie NIST SP 800-90B [6]. Takto navrhnuté zdroje entropie sú následne využívané v entropických generátoroch, ktorých využitie zasahuje do takmer každej oblasti využitia počítačov – od generovania náhodných čísel, cez počítačové simulácie až po generovanie kryptografických kľúčov. Hlavne pri aplikáciach a weboch pracujúcich s citlivými údajmi je nutné, aby boli generované kľúče skutočne náhodné. Preto je nutné poznať zdroje neurčitosti (entropie) v počítačoch a dodržať požiadavky kladené štandardom. Aby bola zaručená spoľahlivosť zdroja entropie počas celej jeho doby života, je nutné ho pravidelne testovať a pred prvotným použitím overiť jeho parametre.

V rámci tejto práce bola popísaná samotná entropia a jednotlivé komponenty zdroja entropie, pričom bol kladený dôraz na popis kľúčovej časti – zdroja šumu. Práve zdroj šumu predstavuje v zdroji entropie pôvodcu „náhody“, preto sa práca zameriava na princípy ich fungovania, ich technické prevedenie, prevádzkové podmienky a možné spôsoby modifikácie či napadnutia.

V druhej časti práce je popísaný proces validácie a hodnotenia zdrojov entropie spolu s potrebnými testami. Tretia časť bola venovaná programom pre získavanie bitových postupností zo zdrojov entropie a programu pre hodnotenie samotných zdrojov entropie.

V závere práce sú porovnané zdroje entropie, z ktorých je možné v rámci navrhnutého programu získavať bitové postupnosti. Je popísaná metodika získavania dát, spôsob ich vyhodnocovania, základné parametre zdrojov a je vytvorená veličina výdatnosť, na základe ktorej sú jednotlivé zdroje porovnávané. Spomedzi hodnotených zdrojov najlepšie obstála zvuková karta (konkrétne zvukový vstup – mikrofón), ktorá predstavuje stabilný zdroj entropie. Nad rámec požiadaviek je ku koncu práce venovaný priestor použitiu hešovacích funkcií v spojení so zdrojmi entropie.

Prínosom tejto diplomovej práce je úvod do problematiky teórie informácie, analýza a podrobný popis zdrojov entropie a metodiky ich testovania. V rámci práce boli vytvorené dva samostatné programy, ktoré umožňujú získanie dát z popisovaných zdrojov a ich následné ohodnotenie. Výstupom práce je porovnanie zdrojov entropie, ktoré vie poslúžiť ako základ pri návrhu alebo analýze entropických generátorov.

Literatúra

[1] HARRISON, R. L., GRANJA, C., LEROY, C. Introduction to Monte Carlo Simulation. In *AIP Conf Proc* [online]. 2010, s. 17–21 [cit. 9. 11. 2019]. DOI: 10.1063/1.3295638. Dostupné z URL: <http://aip.scitation.org/doi/abs/10.1063/1.3295638>.

[2] LAIDLER, K. J. *The World of Physical Chemistry*. Oxford: Oxford University Press, 1993. ISBN 0-19-855919-4.

[3] SHANNON, C. E., WEAVER, W. *The mathematical theory of communication*. Chicago: University of Illionois Press, 1998. ISBN 0252725484.

[4] PROAKIS, J. G. *Digital communications*. 4th ed. Boston: McGraw-Hill, 2001. ISBN 0-07-232111-3.

[5] ŠILHAVÝ, P. *Datová komunikace*. Brno: Vysoké učení technické v Brně, 2012. ISBN 978-80-214-4455-3.

[6] TURAN, M. S., BARKER, E., KELSEY, J. *Recommendation for the Entropy Sources Used for Random Bit Generation* [online]. NIST SP 800-90B. National Institute of Standards and Technology, Gaithersburg, 2018. Dostupné z URL: <https://csrc.nist.gov/publications/detail/sp/800-90b/final>.

[7] KOLÁŘ, M. *Entropický generátor náhodných čísel* [online]. 80 strán. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015 [cit. 11. 11. 2019]. Dostupné z URL: <https://www.vutbr.cz/studium/zaverecne-prace?zp_id=85398>.

[8] BARKER, E., KELSEY, J. *Recommendation for Random Number Generation Using Deterministic Random Bit Generators* [online]. NIST SP 800-90A. National Institute of Standards and Technology, Gaithersburg, 2015. Dostupné z URL: <https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final>.

[9] HU, Y., LIAO, X., WONG, K., ZHOU, Q. A true random number generator based on mouse movement and chaotic cryptography. In *Chaos, Solitons & Fractals* [online]. 2009, 40(5), s. 2286-2293 [cit. 11. 11. 2019]. DOI: 10.1016/j.chaos.2007.10.022. ISSN 09600779. Dostupné z URL: <https://linkinghub.elsevier.com/retrieve/pii/S0960077907008958>.

[10] MARANDI, A., LEINDECKER, N. C., VODOPYANOV, K. L. All-optical quantum random bit generation from intrinsically binary phase of parametric oscillators. In *Optics Express* [online]. 2012, 20(17), [cit. 17. 11. 2019]. DOI: 10.1364/OE.20.019322. ISSN 1094-4087. Dostupné z URL: <https://www.osapublishing.org/abstract.cfm?URI=oe-20-17-19322>.

[11] GABRIEL, CH., WITTMANN, CH., SYCH, D. A generator for unique quantum random numbers based on vacuum states. In *Nature Photonics* [online]. 2010, 4(10), s. 711-715 [cit. 17. 11. 2019]. DOI: 10.1038/nphoton.2010.197. ISSN 1749-4885. Dostupné z URL: <http://www.nature.com/articles/nphoton.2010.197>.

[12] SYMUL, T., ASSAD, S. M., LAM, P. K. Real time demonstration of high bitrate quantum random number generation with coherent laser light. In *Applied Physics Letters* [online]. 2011, 98(23), [cit. 17. 11. 2019]. DOI: 10.1063/1.3597793. ISSN 0003-6951. Dostupné z URL: <http://aip.scitation.org/doi/10.1063/1.3597793>.

[13] NOLL, L. C., COOPER, S. *Lavarand*. 2000, [cit. 17. 11. 2019]. Dostupné z URL: <http://www.lavarand.org/what/how-good.html>.

[14] WALKER, J. *HotBits: Genuine random numbers, generated by radioactive decay*. 1996, [cit. 17. 11. 2019]. Dostupné z URL: <https://www.fourmilab.ch/hotbits/>.

[15] HAAHR, M. *RANDOM.ORG*. 1998, [cit. 17. 11. 2019]. Dostupné z URL: <https://www.random.org/>.

[16] JAMES, F. A review of pseudorandom number generators. In *Computer Physics Communications* [online]. 1990, 60(3), s. 329-344 [cit. 20. 11. 2019]. DOI: 10.1016/0010-4655(90)90032-V. ISSN 00104655. Dostupné z URL: <https://linkinghub.elsevier.com/retrieve/pii/001046559090032V>.

[17] LOVRIC, M. *International encyclopedia of statistical science*. New York: Springer, 2011. Springer reference. ISBN 3-642-04897-8.

[18] Class Random. In *Java Platform Standard Edition 8 Documentation*. [cit. 20. 11. 2019]. Dostupné z URL: <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>.

[19] STERN, J. Secret linear congruential generators are not cryptographically secure. In *Annual Symposium on Foundations of Computer Science (sfcs 1987)*

[online]. IEEE, 1987, s. 421-426 [cit. 20. 11. 2019]. DOI: 10.1109/SFCS.1987.51. ISBN 0-8186-0807-2. Dostupné z URL: <http://ieeexplore.ieee.org/document/4568296/>.

[20] BLACK, P. E. Fisher-Yates shuffle. In *Dictionary of algorithms and data structures*. 2005, 19 [cit. 1. 12. 2019].

[21] SEWARD, J. *bzip2 and libbzip2*. 1996, [cit. 1. 12. 2019]. Dostupné z URL: <http://www.bzip.org>.

[22] ZIV, J., LEMPEL, A. A universal algorithm for sequential data compression. In *IEEE Transactions on Information Theory* [online]. IEEE, 1977, 23(3), s. 337-343 [cit. 3. 2. 2020]. DOI: 10.1109/TIT.1977.1055714. ISSN 0018-9448. Dostupné z URL: <http://ieeexplore.ieee.org/document/1055714/>.

[23] SALOMON, D. *Data Compression* [online]. London: Springer London, 2007 [cit. 3. 2. 2020]. DOI: 10.1007/978-1-84628-603-2. ISBN 978-1-84628-602-5.

[24] HETZEL, W. C., HETZEL, B. *The complete guide to software testing*. Wellesley, MA: QED Information Sciences, 1988. ISBN 0894352423.

[25] TORVALDS, L. *Git*. 2005, [cit. 12. 12. 2019]. Dostupné z URL: <https://git-scm.com/>.

[26] PRESTON-WERNER, T., WANSTRATH, CH., HYETT, P. J. *GitHub*. 2008, [cit. 12. 12. 2019]. Dostupné z URL: <https://github.com/>.

[27] KERNIGHAN, B. W., RITCHIE, D. *The C Programming language*. Englewood Cliffs, N.J.: Prentice-Hall, 1978. ISBN 978-0-13-110163-0.

[28] STROUSTRUP, B. *Standard C++*. 1972, [cit. 12. 12. 2019]. Dostupné z URL: <https://isocpp.org/>.

[29] *Cannonical Ltd. Ubuntu*. 2004, [cit. 12. 12. 2019]. Dostupné z URL: <https://ubuntu.com/>.

[30] BENCINA, R., BURK, P. *PortAudio*. 2011, [cit. 1. 5. 2020]. Dostupné z URL: <http://www.portaudio.com/>.

[31] THE TCPDUMP GROUP *TCPDDUMP & LIBPCAP*. 2020, [cit. 1. 5. 2020]. Dostupné z URL: <https://www.tcpdump.org/>

[32] *X.Org*. 2020, [cit. 1. 5. 2020]. Dostupné z URL: <https://www.x.org/>

[33] KITWARE, INC. *CMake*. 2020, [cit. 1. 5. 2020]. Dostupné z URL: <https://cmake.org/>

[34] GRIESEMER, R., PIKE, R., THOMPSON, K. *The Go Programming Language*. 2009, [cit. 12. 12. 2019]. Dostupné z URL: <https://golang.org/>.

[35] HYKES, S. *Docker*. 2013, [cit. 12. 12. 2019]. Dostupné z URL: <https://www.docker.com/>.

[36] GUTTERMAN, Z., PINKAS, B., REINMAN, T. Analysis of the Linux random number generator. In *2006 IEEE Symposium on Security and Privacy (S&P'06)* [online]. IEEE, 2006, s. 15 pp.-385 [cit. 17. 5. 2020]. DOI: 10.1109/SP.2006.5. ISBN 0-7695-2574-1. Dostupné z URL: <http://ieeexplore.ieee.org/document/1624027/>.

Zoznam symbolov, veličín a skratiek

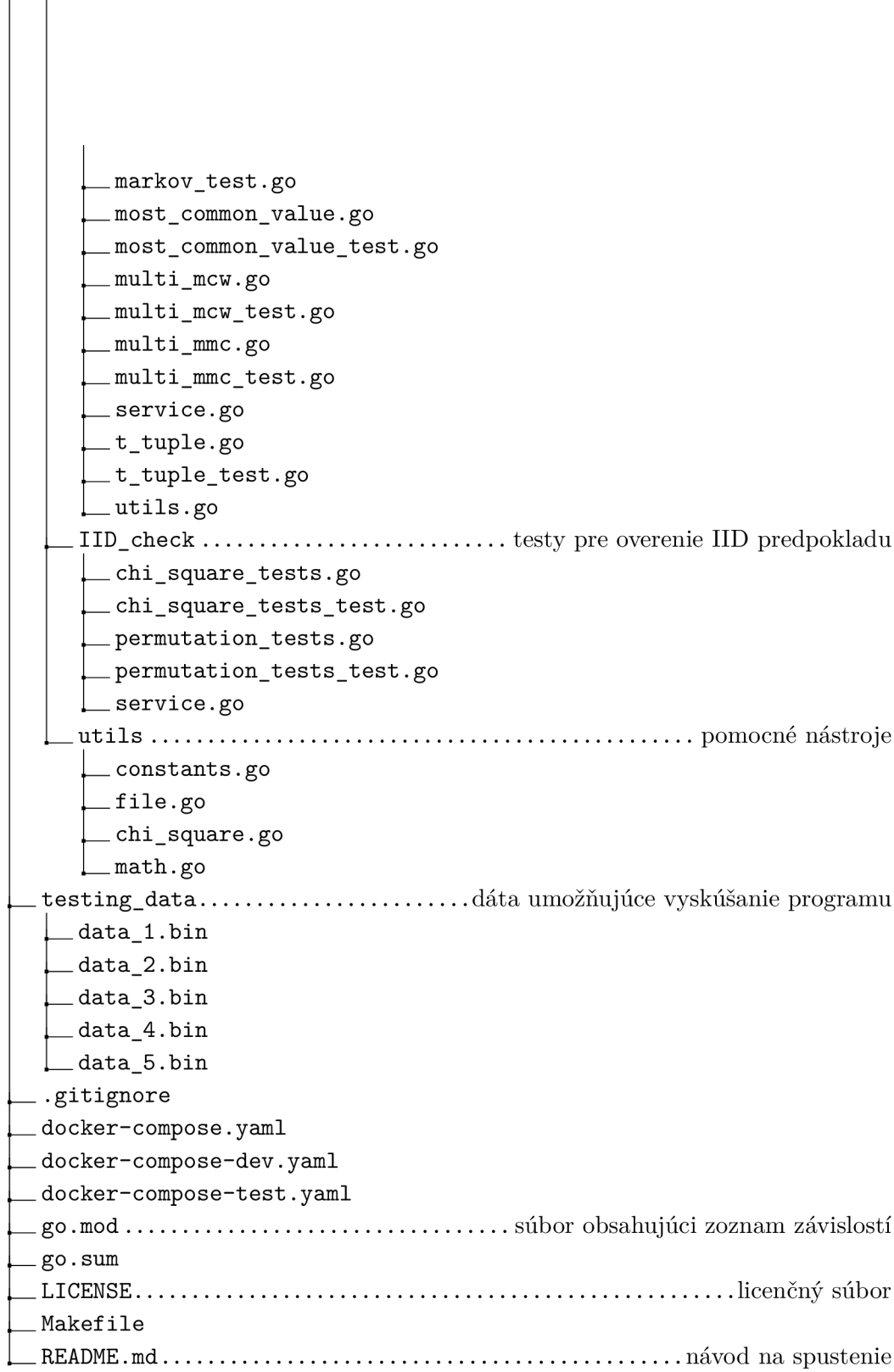
AES	štandard pokročilého šifrovania – *Advanced Encryption Standard*
API	rozhranie pre programovanie aplikácií – *Application Programming Interface*
CBC-MAC	autentizačný kód správy založený na reťazení šifrových blokov – *Cipher Block Chaining Message Authentication Code*
CMAC	autentizačný kód správy založený na šifre – *Cipher-based Message Authentication Code*
DRBG	deterministický generátor náhodných bitov – *Deterministic Random Bit Generator*
HMAC	autentizačný kód správy založený na heši – *Hash-based Message Authentication Code*
HRNG	hardvérový generátor náhodných čísel – *Hardware Random Number Generator*
IID	nezávisle a identicky distribuované – *Independent and Identically Distributed*
LCG	lineárny kongruentný generátor – *Linear Congruential Generator*
LTS	verzia s dlhou podporou – *Long-Term Support*
MD	spracovanie/skrátenie správy – *Message Digest*
NIST	národný inštitút pre štandardy a technológie – *National Institute of Standards and Technology*
NVLAP	národný program dobrovoľnej akreditácie laboratórií – *National Voluntary Laboratory Accreditation Program*
PRNG	generátor pseudonáhodný čísel – *Pseudo-Random Number Generator*
RAM	pamäť s náhodným prístupom – *Random Access Memory*
RBG	generátor náhodných bitov – *Random Bit Generator*
SHA	bezpečný hešovací algoritmus – *Secure Hash Algorithm*
TRNG	generátor skutočne náhodných čísel – *True Random Number Generator*
USB	univerzálna sériová zbernica – *Universal Serial Bus*

Zoznam príloh

A Obsah prílohy

V prílohe je možné nájsť 2 priečinky obsahujúce program pre získavanie bitov zo zdrojov entropie a program pre hodnotenie zdrojov entropie. Programy boli testované na linuxovej distribúcii **Ubuntu** [29] vo verzii **18.04.4 LTS**. Požiadavky a návod na spustenie je možné nájsť v kapitole č. 3.

- `/` ..... koreňový adresár prílohy
  - `entropy_sources` ..... program pre získavanie bitov zo zdrojov entropie
    - `libpcap-1.9.1` ..... knižnica pre spracovanie sieťovej prevádzky
      - `...`
    - `portaudio` ..... knižnica pre spracovanie zvukového vstupu
      - `...`
    - `.gitignore`
    - `audio.cpp`
    - `audio.h`
    - `CMakeLists.txt` ..... súbor programu CMake
    - `LICENSE` ..... licenčný súbor
    - `main.cpp` ..... hlavný súbor programu
    - `network.cpp`
    - `network.h`
    - `README.md` ..... návod na spustenie
    - `screen.cpp`
    - `screen.h`
    - `time_measurement.cpp` ..... metódy na meranie trvania záznamu
    - `time_measurement.h`
  - `es_evaluation` ..... program pre hodnotenie zdrojov entropie
    - `build` ..... priečinok obsahujúci Dockerfile k spusteniu
      - `package`
        - `dev.Dockerfile`
        - `Dockerfile`
    - `cmd`
      - `es_evaluation`
        - `main.go` ..... hlavný súbor programu
    - `internal` ..... priečinok s jadrom programu
      - `entropy_estimation` ..... odhady entropie
        - `collision.go`
        - `collision_test.go`
        - `compression.go`
        - `compression_test.go`
        - `lag.go`
        - `lag_test.go`
        - `lrs.go`
        - `lrs_test.go`
        - `lz78y.go`
        - `lz78y_test.go`
        - `markov.go`


markov_test.go
most_common_value.go
most_common_value_test.go
multi_mcw.go
multi_mcw_test.go
multi_mmc.go
multi_mmc_test.go
service.go
t_tuple.go
t_tuple_test.go
utils.go
IID_check ..... testy pre overenie IID predpokladu
chi_square_tests.go
chi_square_tests_test.go
permutation_tests.go
permutation_tests_test.go
service.go
utils ..... pomocné nástroje
constants.go
file.go
chi_square.go
math.go
testing_data ..... dáta umožňujúce vyskúšanie programu
data_1.bin
data_2.bin
data_3.bin
data_4.bin
data_5.bin
.gitignore
docker-compose.yaml
docker-compose-dev.yaml
docker-compose-test.yaml
go.mod ..... súbor obsahujúci zoznam závislostí
go.sum
LICENSE ..... licenčný súbor
Makefile
README.md ..... návod na spustenie