VÝSLEDEK PROJEKTU STOCHASTICKÉ SYMETRICKÉ BLOKOVÉ ŠIFROVÁNÍ POČÍTAČOVÝCH SOUBORŮ POMOCÍ PROSTŘEDKŮ UMĚLÉ INTELIGENCE PRO ÚČELY CLOUDOVÝCH SLUŽEB - Č. FW01010289

Identifikační číslo:	FW01010289-V1
Název výstupu/výsledku:	Neuro File Encryptor
Druh výsledku:	R – software
Vykazující subjekt:	Vysoké učení technické v Brně
Vlastnické podíly:	Vysoké učení technické v Brně: 50 % Unicorn Systems a.s.: 50 %
Interní registrační číslo výsledku organizace:	167776

Popis výsledku:

Vyvinutý software aplikuje stochastické symetrické blokové šifrování počítačových souborů bajt po bajtu pomocí umělé neuronové sítě adaptované na autoasociativní funkci. Uvedenou adaptací se náhodně vygeneruje šifrovací klíč tvořený konfigurací sítě, tj. synaptickými vahami, který pak bude užit k šifrování, resp. dešifrování počítačového souboru spolu s užitím několika stochastických mechanismů.

1.1 Šifrování souboru užitím neuronových sítí

K šifrování souboru se užije šestnáct neuronových sítí MLP o stejné topologii (pět vrstev o osmi, šestnácti, čtyřech, šestnácti a osmi neuronech), ale různých konfiguracích, tj. synaptických vahách a parametrech sigmoid (prahy a strmosti). Těchto šestnáct sítí resp. jejich konfigurací, tj. klíčů, se generuje v samostatné aplikaci (adaptivní mód sítě) na tréninkové množině obsahující všechny kombinace nastavení osmi bitů, přiváděné během učení sítě současně na vstupní i výstupní vrstvu, tj. tréninková množina obsahuje 256 prvků. Učení každé z šestnácti sítí se zahajuje s jinou výchozí náhodně inicializovanou konfigurací.

Proces šifrování daného souboru probíhá bajt po bajtu v další samostatné aplikaci (aktivní mód sítě), kde zašifrovanou formou každého bajtu jsou stavy neuronů dělící vrstvy sítě (střední vrstva MLP o čtyřech neuronech), tj. čtyři reálná čísla, tj. šestnáct bajtů. Těchto šestnáct bajtů je ještě doplněno o další čtyři bajty, v kterých náhodně plave tzv. řídící bajt obsahující „návod“ pro dešifrování uvedených šestnácti bajtů. Takže obraz každého bajtu původního souboru je představován dvaceti bajty zašifrovaného souboru. Obě aplikace (adaptivní a aktivní mód sítě) jsou programovány v jazyce Fortran.

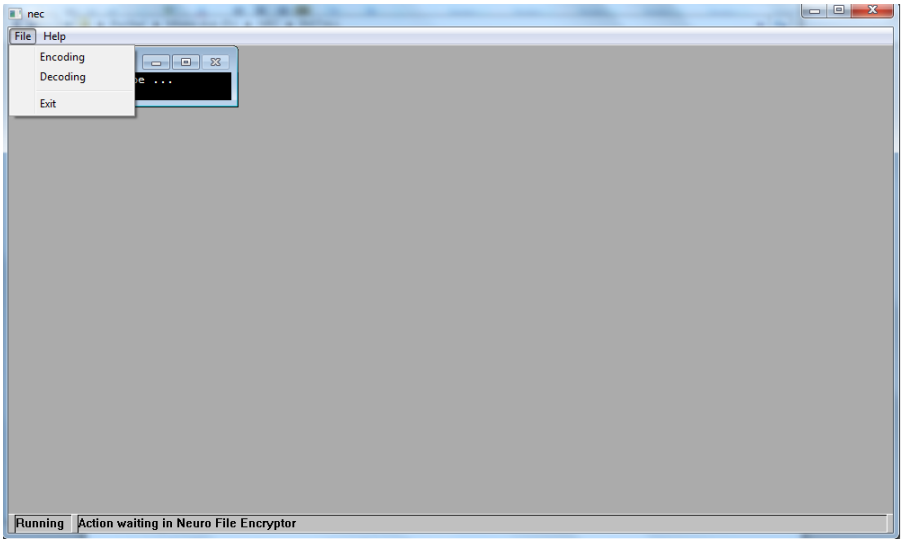
Šifrovací klíč je konfigurace neuronové sítě, tj. všechny synaptické váhy sítě a parametry aktivačních funkcí neuronů skrytých vrstev sítě, tj. strmosti a prahy sigmoid. Konfigurace sítě je

výsledkem adaptace sítě na autoasociativní funkci metodou backpropagation na tréninkové množině obsahující rozšířenou ASCI tabulku v binární formě, tj. obsahující dvě na osmou prvků.

Velikost šifrovacího klíče je dána velikostí celé konfigurace neuronové sítě, tj. 8x16+16x4+4x16+16x8=384 vah, tj. 1536 bajtů, přičemž při šifrování jednoho souboru se náhodně střídá šestnáct konfigurací, resp. klíčů, tj. 24576 bajtů. K tomu se ještě připočte 16x2x36=1152 parametrů sigmoid, tj. 4608 bajtů.

1.2 Integrace funkcí do softwarového rozhraní

Bylo dokončeno uživatelské softwarové rozhraní umožňující spouštění šifrování a dešifrování počítačového souboru (viz Obr. 1).



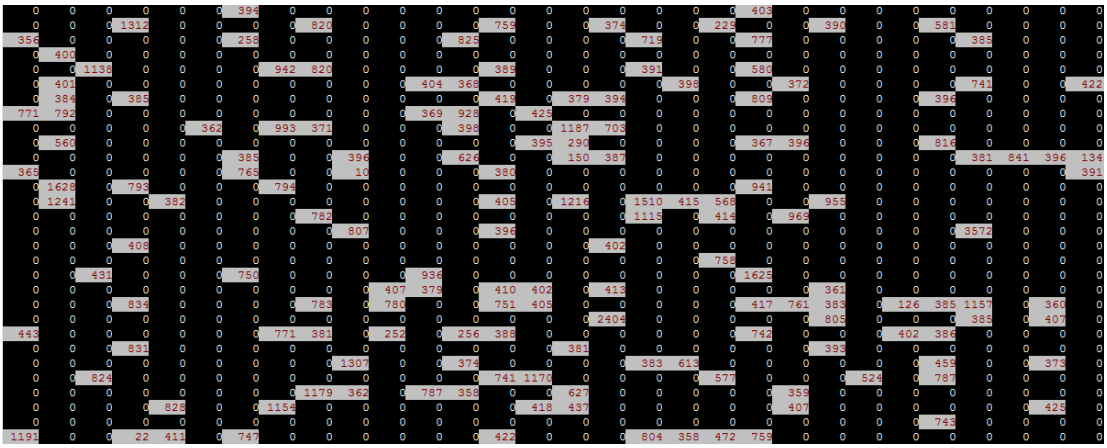
Obr. 1– Hlavní obrazovka SW rozhraní

1.3 Testování integrovaného softwaru

Dále je uveden příklad šifrované podoby jednoho a téhož bajtu (písmeno A) opakovaně po sobě sto tisíc krát zašifrovaného, následovaný ilustrací jeho rozmístění v čtyřrozměrném prostoru užitím Kohonenovy mapy (viz Obr. 2).

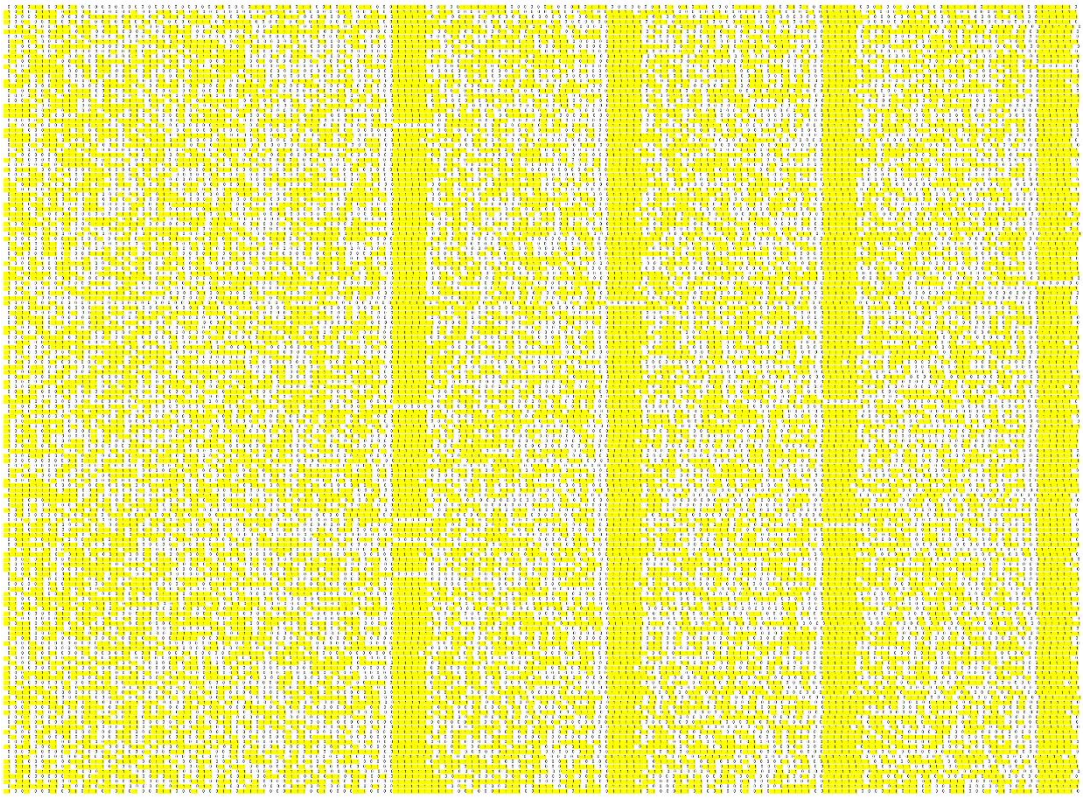
Jako optimální architektura neuronové sítě byla zvolena síť o třech skrytých vrstvách, kde střední skrytá (dělící) vrstva je tvořena čtyřmi neurony, tj. kompresní poměr je zvolen osm (počet vstupních/výstupních neuronů) ku čtyřem, zbylé skryté vrstvy mají po šestnácti neuronech.

Pro šifrování bajtů bylo užito šestnáct konfigurací sítě, získaných adaptací sítě na autoasociativní funkci vždy s jinou počáteční hodnotou sekvence pseudonáhodných čísel, užitou při počáteční náhodné inicializaci každé konfigurace sítě, spolu s několika stochastickými mechanismy zajišťující jinakost každého šifrovaného bajtu.



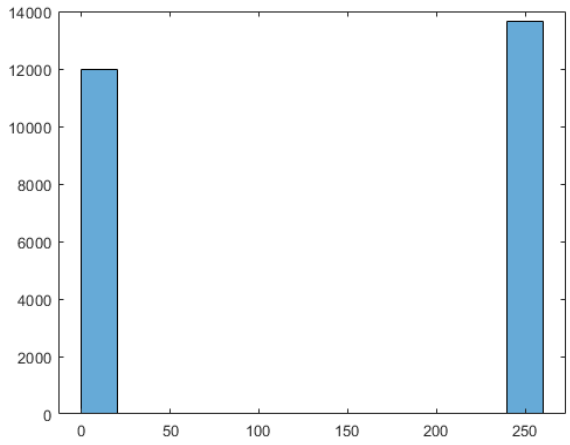
Obr. 2 – Kohonenova mapa

V rámci ověření bezpečnosti vlastního softwarového řešení bylo přistoupeno ke standardizovanému testování vygenerovaných šifrovaných posloupností pomocí vizuálních testů a vybraných statistických testů. Jako první byl vybrán analytický test SVA (Simple Visual Analysis), který vizualizuje generovanou posloupnost do vybraného rozlišení dvou barevného obrazu. Tento test využívá vrozené lidské vlastnosti umožňující snadno rozpoznávat vzory, což je v tomto případě více než žádoucí. Nejedná se o vyčerpávající způsob určení náhodnosti, nicméně o vstupní a velmi rychlý test, který dokáže odhalit případné nedokonalosti generátoru. Rozlišení bylo zvoleno 160x160 bitů, kde šířka odpovídá jedné iteraci a výška je volena dle šířky. Obraz tak reprezentuje 25.6 kb dat. Níže již uvádíme vytvořenou grafickou reprezentaci vygenerovaných dat.



Obr. 3 – Vygenerovaná posloupnost 160x160 bitů a vykreslena pomocí logiky testu SVA

Z obrázku vyplývají drobné nedokonalosti, které vytváří jasně viditelné vzory. Jedná se o vzory tvořené bity exponenciální části bitové reprezentace desetinných čísel, které představují parametry čtyř neuronů dělící vrstvy sítě. I přes značnou míru těchto bitů je stále viditelný šum v ostatních částech vygenerované posloupnosti. Ve spojitosti se značnou mírou naddimenzování (desetinásobek vygenerovaných bitů oproti vstupní posloupnosti) lze tvrdit, že generovaná míra náhodnosti a počtu bitů je dostatečná. V rámci testování bylo dále přistoupeno také k jednoduché matematické metodě, a to srovnání počtu jedniček a nul v rámci dané vygenerované posloupnosti na obrázku. V rámci grafu níže odpovídá hodnota „0“ bitu 0 a hodnota „250“ pak bitu 1.



Obr. 4 – Matematické srovnání výskytu jedniček a nul ve vygenerované posloupnosti.

Tento test ukázal zhruba 10% nesrovnalost mezi 1 a 0 ve vygenerované posloupnosti, což je zapříčiněno převážně vertikálními vzory, které se objevily již při vizuálním testu. Pro hlubší analýzu vytvořeného softwaru bylo následně přistoupeno ke standardizovaným statistickým testům, které jsou běžně používány v rámci baterií NIST STS, DIEHARD či TestU01. Tyto testy využívají vyvrácení nulové hypotézy H_0 (posloupnost je náhodná) s možnou chybou prvního a druhého typu (falešně pozitivní či falešně negativní výsledek). Z tohoto pohledu se využívá tzv. důvěryhodnosti výsledku, který byl pro všechny testy zvolen jako $\alpha = 0.005$ tj. výsledky mají 99.5% důvěryhodnost (či chyba nastane v 0.5 % případů). Pro prvotní testování bylo zvoleno pět primárních testů: frekvenční test, kumulativní dopředný test, kumulativní zpětný test, test nejdelší sekvence a rank test. Testy byly provedeny na jednom miliónu vygenerovaných bitů.

Frekvenční test hodnotí podíl nul a jedniček pro celou sekvenci. Je velice podobný předchozí matematické hodnotě, jen je do tohoto testu zahrnuta již statistická chyba a tolerance pro náhodnost sekvence.

Kumulativní testy obdobné testy z pohledu frekvenčního testu, samotné kroky jsou však v tomto případě náhodné (dopředné či zpětné). Opět se hodnotí celá posloupnost.

Test nejdelší sekvence jedná se o toleranci největšího souběhu jedniček či nul dle distribuce χ^2.

Rank test velice obdobný předchozímu testu, kde se v jednoduchosti testují shluky v rámci sub-matic, opět dle distribuce χ^2.

Test	Výsledek	Popis
#1	$\sum$ = -30230	Výsledek potvrdil nedokonalost v rámci patrných vzorů generátoru i následnou matematickou metodu, kdy není v rámci generované posloupnosti stejný počet nul a jedniček. Chyba v tomto případě je ca 6%. Frekvenční i kumulativní testy vykazují obdobné výsledky, díky velmi podobné logice.
#2	$\sum$ = 30230	
#3	$\sum$ = 30230	
#4	χ^2 = 29363.854628	Tento test potvrzuje zjištění viditelných liniových vzorů ve vygenerované posloupnosti.
#5	χ^2 = 3163.718414	Tento test potvrzuje již viditelné maticové vzory ve vygenerované posloupnosti.

I přes zjištěné nedokonalosti vytvořeného softwaru lze však dedukovat z provedených testů, že samotný algoritmus může být bez jakéhokoliv problému použit v běžných aplikacích, kde poskytne dostatečnou bezpečnost a alternativu k tradičním metodám. Pro kritické aplikace jej však nelze, díky zjištěným nedokonalostem, doporučit (mj. i díky značnému naddimenzování náhodných bitů). Využití v kritických aplikacích využívaných např. v kritické infrastruktuře vyžaduje nadstandardní zabezpečení a jinou úroveň bezpečnosti než běžné aplikace. V tomto případě by bylo nutno provést navazující výzkum, protože splnění těchto nadstandardních parametrů je nad rámec tohoto projektu.

Technické parametry:

64-bitová aplikace NFE spustitelná pod operačním systémem Windows verze 7 a výše. Kódování aplikace probíhalo ve vývojovém prostředí Microsoft Visual Studio, kódování aktivní dynamiky neuronové sítě proběhlo v jazyce Fortran, vhodném pro kódování vědecko technických výpočtů. Pro tvorbu uživatelského rozhraní byla užita vývojová platforma Fortran QuickWin.

Ekonomické parametry:

Výsledek poskytuje efektivní alternativu k běžně užívaným šifrovacím algoritmům (např. AES či RSA), netrpící některými riziky prolomení šifry (objev rychlého algoritmu faktorizace či rozvoj kvantových počítačů), neboť šifrování pomocí umělé inteligence je založené na zcela jiných principech, resp. jeho licenční poplatky mohou vést k úspoře provozních nákladů.

Licence a využití:

K využití výsledku jiným subjektem je vždy nutné nabytí licence. Poskytovatel licence na výsledek požaduje licenční poplatek. Více informací na toman@vut.cz