EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/I/2020/421000013833

POROVNANIE ŠIFROVACÍCH METÓD A ICH EFEKTIVITY V KONKRÉTNYCH PRÍPADOCH

Označenie práce: Diplomová práca

Bratislava 2020 Bc. Jozef Kilik

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

POROVNANIE ŠIFROVACÍCH METÓD A ICH EFEKTIVITY V KONKRÉTNYCH PRÍPADOCH

Označenie práce: Diplomová práca

Študijný program: Informačný manažment

Študijný odbor: Ekonómia a manažment

Školiace pracovisko: Katedra aplikovanej informatiky

Vedúci záverečnej práce: RNDr. Eva Rakovská, PhD.

Bratislava 2020 Bc. Jozef Kilik

Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Jozef Kilik
Študijný program: informačný manažment (Jednoodborové štúdium, inžiniersky II. st., denná forma)
Študijný odbor: ekonómia a manažment
Typ záverečnej práce: Inžinierska záverečná práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Porovnanie šifrovacích metód a ich efektivity v konkrétnych prípadoch.

Anotácia: Autor v práci vyberie vhodné prípadové úlohy vhodné na šifrovanie. V praktickej časti sa analyzujú možnosti šifrovania úloh viacerými metódami, následne sa úlohy zašifrujú vybranými metódami a konkrétne sa porovnajú atribúty šifrovacích metód pre konkrétne úlohy.

Vedúci: RNDr. Eva Rakovská, PhD.
Katedra: KAI FHI - Katedra aplikovanej informatiky FHI
Vedúci katedry: Ing. Mgr. Peter Schmidt, PhD.

Dátum zadania: 24.10.2018

Dátum schválenia: 25.10.2018

Ing. Mgr. Peter Schmidt, PhD.
vedúci katedry

Čestné vyhlásenie

Čestne vyhlasujem, že diplomovú prácu som vypracoval samostatne, na základe konzultácií a v práci som na príslušnom mieste uviedol zoznam všetkej použitej literatúry.

Dátum: 1.4.2020 **.....**

meno autora

Poďakovanie

Týmto by som chcel poďakovať pani RNDr. Eve Rakovskej, PhD., za jej rady, pripomienky a odbornú pomoc pri mojej diplomovej práci.

ABSTRAKT

KILIK, Jozef**:** Porovnanie šifrovacích metód a ich efektivity v konkrétnych prípadoch. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky: Katedra aplikovanej informatiky. – Vedúci záverečnej práce: RNDr. Eva Rakovská, PhD. – Bratislava: FHI EU, 2020, (81 s.)

Cieľom diplomovej práce je porovnanie šifrovacích metód a ich efektivity v konkrétnych prípadoch. Na splnenie cieľa je dôležité preštudovať a vysvetliť jednotlivé šifrovacie algoritmy, popísať niektoré šifrovacie softvéry a vybrať vhodné prípadové úlohy vyhovujúce na šifrovanie. Práca je rozdelená do troch hlavných kapitol. Obsahuje 3 tabuľky. Prvá kapitola sa zaoberá kryptografiou, popisom šifrovacích algoritmov, ktoré sa budú porovnávať, a priblížením šifrovacích softvérov, v ktorých sa budú jednotlivé prípadové úlohy šifrovať. V ďalšej kapitole sa podrobne rozpisuje hlavný cieľ práce s jeho čiastkovými cieľmi, ktoré musia byť splnené. Posledná kapitola sa venuje interpretovaniu výsledkov práce. Prípadové úlohy sa zašifrujú vybranými metódami a atribúty šifrovacích metód sa porovnajú pre konkrétne úlohy. Výsledkom riešenia je nájdenie najefektívnejšieho šifrovacieho softvéru pre šifrovanie algoritmu AES a zároveň porovnanie efektívnosti dvoch algoritmov v softvéri Cryptool.

Kľúčové slová:

Kryptografia, šifrovací algoritmus, šifrovací softvér, efektivita

ABSTRACT

KILIK, Jozef: Encryption methods comparison and their effectivity on concrete cases. - University of Economics in Bratislava. Faculty of Economic Informatics: Department of Applied Informatics. - Supervisor: RNDr. Eva Rakovská, PhD. - Bratislava: FHI EU, 2020, (81 pages)

The aim of the thesis is to compare encryption methods and their effectiveness in specific cases. To accomplish this goal, it is important to study and explain the individual cryptographic algorithms, to describe some of the cryptographic software, and to select the appropriate case tasks suitable for encryption. The thesis consists of three main chapters. The first chapter deals with cryptography, the description of encryption algorithms to be compared and the approach of encryption software, in which the individual case tasks will be encrypted. The next chapter details the main objective of the work with its partial objectives that must be met. The last chapter is devoted to interpreting the results of the work. Case tasks are encrypted by selected methods, and the attributes of the encryption methods are compared for specific tasks. The result of the solution is to find the most effective encryption software for encryption of the AES algorithm and at the same time to compare the efficiency of two algorithms in Cryptool software.

Keywords:

Cryptography, encryption algorithm, encryption software, efficiency

OBSAH

Úvod

V súčasnosti je šifrovanie veľmi dôležité, pretože chráni informácie takmer bez toho, aby sme si to vôbec uvedomovali. Bez kvalitného šifrovania by nemohli existovať rôzne najmodernejšie technológie, ako napríklad online nakupovanie a internet banking. Šifrovanie je dôležité aj pre obyčajných ľudí, ktorí si potrebujú zabezpečiť svoje súbory pred nebezpečenstvom tretích strán. Na tieto účely sú vytvorené mnohé šifrovacie softvéry, ktoré umožňujú šifrovanie pomocou najmodernejších a najbezpečnejších šifrovacích algoritmov. Všade okolo nás vo svete IT existuje mnoho hrozieb, ktoré čakajú, kým prestaneme byť dostatočne obozretní a zaútočia práve vtedy, keď je naša ochrana údajov najzraniteľnejšia. Preto je potrebné mať dobre zabezpečené všetky dôležité údaje, ktoré môžu byť zneužité, práve šifrovacími softvérmi. V tejto záverečnej práci zameriame svoju pozornosť na uvedenú problematiku a budeme porovnávať viaceré takéto softvéry so šifrovacím algoritmom AES a zisťovať, ktorý najefektívnejšie šifruje prípadové úlohy. Taktiež zhodnotíme rozdiel efektivity šifrovania v programe Cryptool medzi algoritmom AES a RSA.

Diplomová práca sa pozostáva z troch hlavných kapitol. Prvá z nich sa venuje súčasnej situácii v kryptografii, ako aj jej popisom a všeobecným rozdelením jednotlivých šifrovacích algoritmov. Okrem ich rozdelení sa táto kapitola zaoberá aj bližším popisom vybraných šifrovacích algoritmov, z ktorých sa algoritmy AES a RSA použijú v praktickej časti. Ich popis a analýza je dôležitým čiastkovým cieľom, ktorý je potrebné pre naplnenie hlavného cieľa tejto práce. Ďalším čiastkovým cieľom v tejto kapitole, je oboznámenie sa s dostupnými bezplatnými šifrovacími softvérmi a šifrovacími aplikáciami, následne ich priblížením a vysvetlením ich podstaty fungovania. Na základe získaných vedomostí sa použijú niektoré šifrovacie softvéry (Cryptool, AxCrypt, 7-Zip File Manager a Secure IT), v ktorých sa otestuje ich efektivita šifrovania konkrétnych algoritmov.

Druhá kapitola opisuje hlavný cieľ a metodiku tejto diplomovej práce. Rozpisuje podrobne, aké čiastkové ciele musia byť splnené, aby sa mohol tento celkový cieľ naplniť. Taktiež približuje metodiku, ktorá sa používa pri spĺňaní týchto cieľov a zároveň popisuje odkiaľ boli čerpané zdroje na celkovú tvorbu záverečnej práce.

Posledná hlavná kapitola tejto práce je zameraná na praktickú časť, kde prichádza ku konkrétnemu postupnému vysvetľovaniu zistených výsledkov. Jednotlivé prípadové úlohy

sa tu šifrujú vo vybraných šifrovacích softvéroch prostredníctvom algoritmov AES a RSA. Sú tu popísané jednotlivé kroky, ktoré sa vykonávali v softvéroch, aby bolo umožnené zašifrovať dané prípadové úlohy. Po získaní výsledkov zo šifrovania prípadových úloh sa porovnajú viaceré parametre, na základe ktorých sa určí celková efektivita jednotlivých šifrovacích softvérov. Tieto parametre sú čas, za ktorý proces šifrovania prebehne, celková veľkosť pôvodného a zašifrovaného súboru, ako aj veľkosť, ktorú zaberajú na disku počítača, a posledným parametrom je čas odšifrovania. Pri porovnaní všetkých týchto parametrov šifrovania prípadových úloh algoritmom AES sa určí hlavný cieľ tejto práce - najefektívnejší softvér. Ďalším cieľom, ktorý sa zistí, je nájdenie vhodnejšieho šifrovania prostredníctvom algoritmov AES a RSA v softvéri Cryptool.

1 Súčasný stav riešenej problematiky

V tejto kapitole sa budeme zaoberať pojmom kryptografia a pojmami s ním spojených. Popíšeme si význam kryptografie, jednotlivé šifrovacie algoritmy, ktoré sa používali v priebehu času a ktoré sú najvyužívanejšie a najrozšírenejšie v súčasnosti.

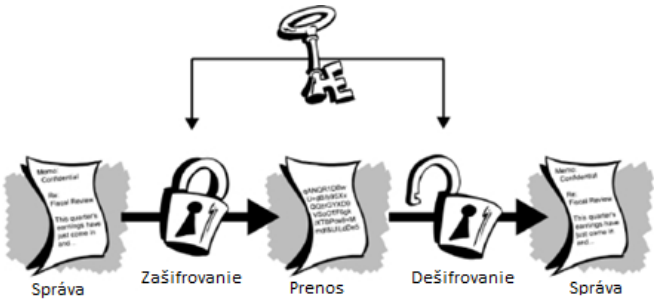
1.1 Kryptografia

Kryptografia zahŕňa vytváranie generovaných alebo písaných kódov, ktoré potom umožňujú utajenie informácií. Kryptografia potrebné údaje transformuje do formátu, ktorý nie je čitateľný pre neautorizovaného používateľa, čo znamená, že jeho prenos sa vykoná bez toho, aby ho neautorizované subjekty dekódovali späť do čitateľného formátu, čím by dôšlo k ohrozeniu údajov. Informačná bezpečnosť používa kryptografiu na viacerých úrovniach. Informácie sa nedajú prečítať bez toho, aby sa použil kľúč na ich dešifrovanie. Tieto zašifrované informácie si zachovávajú svoju integritu počas prepravy a počas ukladania. Kryptografia tiež pomáha pri tom, že správy odosielateľa a doručenie môžu byť overené. [1]

Možností, ako zašifrovať nejakú správu tak, aby sme neautorizovanému subjektu sťažili prístup čo najviac, je veľmi veľa. Z praktického hľadiska sa dá uvažovať o nemožnosti dešifrovania správy. Inak povedané, využitím najmodernejších rýchlych počítačov je to takmer nemožné (trvalo by to niekoľko rokov). Dešifrovanie sa spája s problémom, ktorý sa dá chápať aj ako problém zistenia kľúča, ktorým sa správa zašifrovala. Podľa kľúča, ktorý sa využíva na zašifrovanie hovoríme o symetrickej a asymetrickej kryptografii. [2]

1.1.1 Symetrická kryptografia

Symetrická kryptografia je typ šifrovania, pri ktorom sa na šifrovanie aj dešifrovanie elektronických informácií využíva iba jeden kľúč (tajný kľúč). Subjekty, ktoré komunikujú prostredníctvom tohto šifrovania, si musia tento tajný kľúč vymeniť, aby sa dal použiť pri dešifrovaní. Tento druh šifrovania sa líši od asymetrickej kryptografii tým, že na šifrovanie a dešifrovanie správ sa používajú dva kľúče - jeden kľúč je verejný a jeden súkromný. [3]



Obrázok 1 Symetrická kryptografia [33]

Poznáme dva typy symetrických šifrovacích algoritmov podľa toho, ako spracovávajú text :

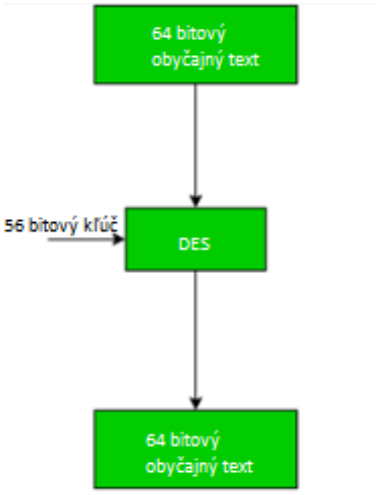
- Blokové algoritmy - nastavené dĺžky bitov sú šifrované v blokoch elektronických údajov pomocou špecifického tajného kľúča. Keď sú dáta šifrované, systém uchováva údaje vo svojej pamäti, keď čaká na úplné bloky. Medzi blokové algoritmy patria DES, RC2, 3DES, AES, Blowfish, Camelia, Serpent a Twofish. [4]

- Prúdové algoritmy - dáta sú šifrované, pretože sa prenášajú, namiesto toho, aby sa uchovávali v pamäti systému. Užitočné sú preto, lebo znaky sa šifrujú oddelene, čo znamená, že sa využívajú v prostredí, kde nastávajú veľké straty pri prenose. Medzi prúdové algoritmy patria OTP, RC4, Salsa20 a CSS. [4]

DES

Data encryption standart (DES) bol označený ako zraniteľný voči útokom hrubou silou, a preto sa popularita DES mierne znížila. [5] Federálna vláda USA ho vyvinula, aby poskytla kryptografickú bezpečnosť pre všetky vládne komunikácie. [6]

DES je bloková šifra, ktorá šifruje údaje v blokoch s veľkosťou 64 bitov, čo znamená, že vstupom do DES je 64 bitov obyčajného textu, z ktorého sa vytvára 64 bitová šifra. Rovnaký algoritmus a kľúč sa používajú na šifrovanie a dešifrovanie s malými rozdielmi. Dĺžka kľúča je 56 bitov. [5]



Obrázok 2 DES princíp [5]

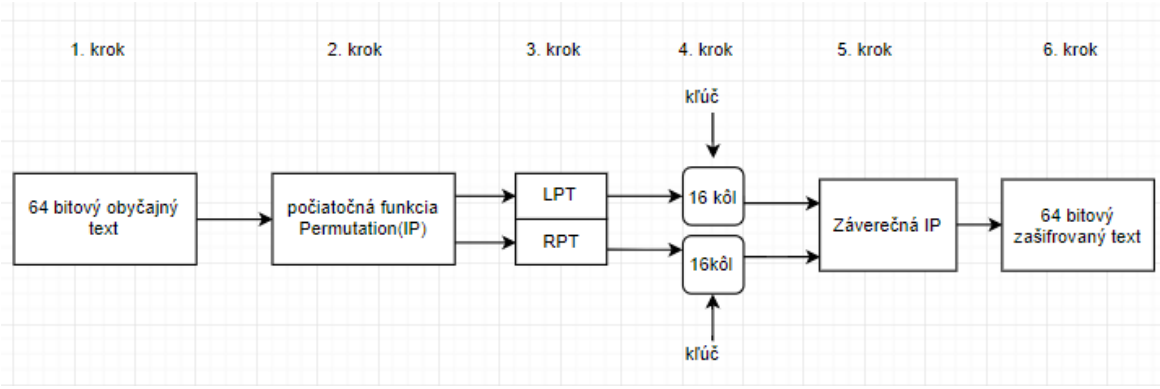
Ako je vyššie spomenuté, DES pri šifrovaní využíva 56-bitový kľúč. Začiatočný kľúč sa však v skutočnosti skladá zo 64 bitov. Pred začatím šifrovanie pomocou DES sa každý 8. bit kľúča vynechá, čím sa vytvorí 56 bitový kľúč. Vynechá sa teda bitová pozícia 8, 16, 24, 32, 40, 48, 56 a 64.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Obrázok 3 DES tabuľka [5]

DES je postavený na dvoch základných metódach kryptografie: substitúcia (tiež zámena) a transpozícia (tiež difúzia). DES pozostáva zo 16 krokov, z ktorých sa každý jeden krok nazýva kolo. V každom kole sa vykonávajú kroky substitúcie a transpozície.

1. V prvom kroku sa 64-bitový obyčajný textový blok odovzdá počiatočnej funkcii Permutation (IP).
2. Táto počiatočná funkcia (IP) sa vykonáva na obyčajnom texte.
3. IP rozdelí text na dve polovice permutovaného bloku; ľavý obyčajný text (LPT) a pravý obyčajný text (RPT).
4. Teraz každý LPT a RPT prejde 16 kolami šifrovacieho procesu.
5. Nakoniec sa LPT a RPT znovu spoja a na kombinovanom bloku sa vykoná záverečná permutácia (FP)
6. Výsledkom tohto procesu je 64 bitový šifrovací text. [5]



Obrázok 4 DES popis [6]

V súčasnosti sa DES nevyužíva, lebo bol oficiálne stiahnutý v roku 2005. [29]

3DES

Triple data encryption algorithm (3DES) bol jednou z najvýznamnejších foriem šifrovania, ktorá bola založená na algoritme DES. Postupne ho nahradil vo väčšine prípadov algoritmus AES. [6]

3DES je šifrovací algoritmus, ktorý bol odvodený zo štandardu Data Encryption Standard (DES). Na konci deväťdesiatych rokov sa stal popredným, ale od tej doby začal postupne upadať z dôvodu zvýšenia bezpečnejších algoritmov. Aj keď je už zastaraný, v niektorých situáciách sa stále používa. Hoci je oficiálne známy ako Triple data encryption algorithm (3DEA), označuje sa ako 3DES. Je to preto, lebo algoritmus 3DES používa šifru Data Encryption Standard (DES) trikrát na šifrovanie údajov. DES je algoritmus so symetrickým kľúčom, čo znamená, že šifra používa rovnaký kľúč pre šifrovacie aj dešifrovacie procesy. Ako sme sa už dozvedeli, DES má 64-bitový blok aj veľkosť kľúča, ale v praxi poskytuje kľúč iba 56-bitovú bezpečnosť. 3DES bol preto vyvinutý ako tá bezpečnejšia alternatíva z dôvodu krátkej dĺžky kľúča DES. V 3DES sa algoritmus DES vykonáva trikrát prostredníctvom troch kľúčov. Za bezpečný sa však považuje len vtedy, ak sa použijú tri samostatné kľúče. [6]

Po zistení slabín normálneho DES, bol 3DES prijatý vo veľkom množstve aplikácií. Bol to jeden z najbežnejších používaných šifrovacích algoritmov pred vznikom AES. Medzi príklady jeho používania patrili napríklad platobné systémy, Microsoft Office,

Firefox, EMV a ďalšie. Viacero z týchto platforiem už 3DES nevyužíva, pretože už existujú lepšie alternatívy.

Keď sa nedostatky bezpečnosti DES stali viditeľné, 3DES bol navrhnutý ako rozšírenie jeho veľkosti kľúča bez toho, aby sa musel budovať úplne nový algoritmus. Teda namiesto použitia iba jedného kľúča ako v DES, 3DES vykonáva algoritmus DES trikrát s tromi 56-bitovými kľúčmi:

1. kľúč sa používa na šifrovanie obyčajného textu.
2. kľúč sa používa pri dešifrovaní textu, ktorý bol zašifrovaný 1. kľúčom.
3. kľúč sa používa na šifrovanie textu, ktorý bol dešifrovaný 2. kľúčom. [6]

Akonáhle 2. kľúč „dešifruje“ text, použije sa 3. kľúč na jeho opätovné zašifrovanie. Výsledkom je šifra 3DES.

Technicky sa 3DES môže implementovať s tromi odlišnými konfiguráciami kľúčov. Z toho je však druhá a tretia možnosť neistá a nemala by sa nikdy implementovať.

- 1 možnosť - tri nezávislé kľúče a je najbezpečnejšia.
- 2 možnosť - prvý a tretí kľúč sú rovnaké.
- 3 možnosť - tri identické kľúče. Výsledok je rovnaký ako pri bežných DES.[6]

Spoločnosťou NIST bolo 19. júla 2018 uverejnené, že sa algoritmus Triple Data Encryption Algorithm oficiálne vyraďuje. Po období verejných konzultácií sa 3DES zastaví pre všetky nové aplikácie a jeho používanie sa po roku 2023 nepovolí. 3DES je však hlavný algoritmus zabudovaný v platobných systémoch, štandardoch a technológií vo finančnom priemysle. Návrh NIST na päťročný časový plán zakázania používania systému 3DES môže pre tento priemysel predstavovať veľké problémy z dôvodu nevykonateľnej infraštruktúry, kreditných kariet v obehu a možných problémov s interoperabilitou. [6]

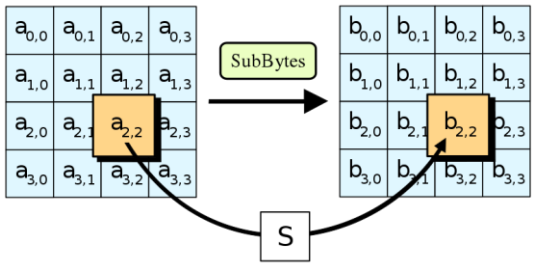
AES

Advanced Encryption standard, inak nazvaný aj Rijndael, patrí medzí symetrické algoritmy, čo znamená, že používa jeden kľúč, ktorý slúži na zašifrovanie aj dešifrovanie dát. Dĺžka tohto kľúča je 128, 196 alebo 256 bitov. Šifra AES je zabezpečená oproti útoku hrubou silou, čiže vyskúšaniu všetkých možných kľúčov, ako napríklad u DES, ktoré je možné prelomiť. V súčasnej dobe je nemožný útok hrubou silou ani na 128-bitový kľúč.

Jediná potenciálna hrozba útoku hrubou silou na túto šifru je zo strany kvantových počítačov a počítačov, ktoré sú založené na báze DNA. AES je blokový šifrovací algoritmus, ktorý sa aplikuje na dáta s pevne stanovenou dĺžkou - v tomto prípade 128 bitov. Keď sú šifrované dáta dlhšie, ich spracovávanie sa vykonáva po jednotlivých blokoch. Ak sú však dáta kratšie, je potrebné ich doplniť, aby zodpovedali stanovenej dĺžke. Toto doplnenie na stanovenú dĺžku sa nazýva "padding". Existujú pre neho mnoho algoritmov, od jednoduchého doplnenia núl po zložitejšie algoritmy doplnenia.

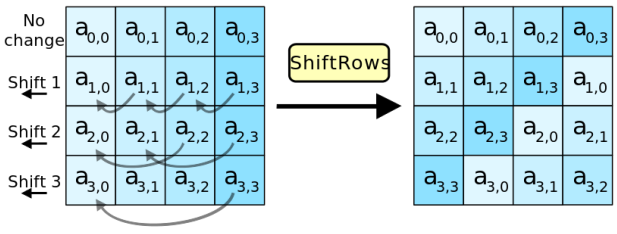
Šifrovanie pomocou AES prebieha v 4 krokoch.

1. SubBytes - jednoduchá substitúcia, v ktorej je každý jeden bajt nahradený iným podľa stanoveného kľúča (8-bitového vymieňacieho boxu). Touto operáciou sa zabezpečuje nelineárnosť šifry a bráni útokom, ktoré sa zakladajú na jednoduchých algebrických vlastnostiach. [7]



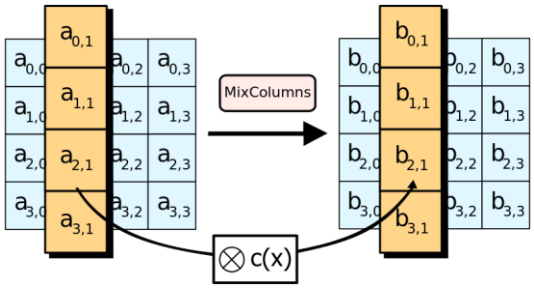
Obrázok 5 Substitúcia [7]

2. ShiftRows – v tomto kroku sa jednotlivé bajty prehadzujú podľa úrovní riadkov. V prvom riadku sa nič nemení a všetky bajty ostávajú na pôvodnom mieste. V druhom riadku sa bude každý bajt presúvať smerom doľava o jeden bajt. Tretí riadok sa bude posúvať o dva miesta doľava a štvrtý o tri miesta vľavo. Tento krok je dôležitý, pretože sa stĺpce nezávisle šifrujú . [7]

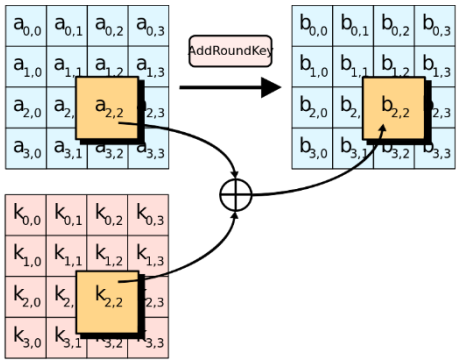


Obrázok 6 Presun bajtov[7]

3. MixColumns – v tomto kroku dochádza k poprehadzovaniu stĺpcov a zároveň sa každý stĺpec vynásobí rovnakým polynómom c (x). Týmto krokom získava šifra dostatočnú náhodnosť. [7]



Obrázok 7 Prehadzovanie stĺpcov[7]

4. AddRoundKey — v poslednom kroku je každý bajt skombinovaný s podkľúčom, ktorý získame z pôvodného kľúča za pomoci Rijndaelovej tabuľky. Všetky bajty podkľúča potom skombinujeme so svojim príslušným bajtom nášho textu a dostaneme výslednú šifru.[7]
5.
6. 

Obrázok 8 Pridanie podkľúča[7]

V súčasnosti je 256-bitový AES štandardným algoritmom pre banky, ktoré ho využívajú na ochranu dát pred tretími stranami.

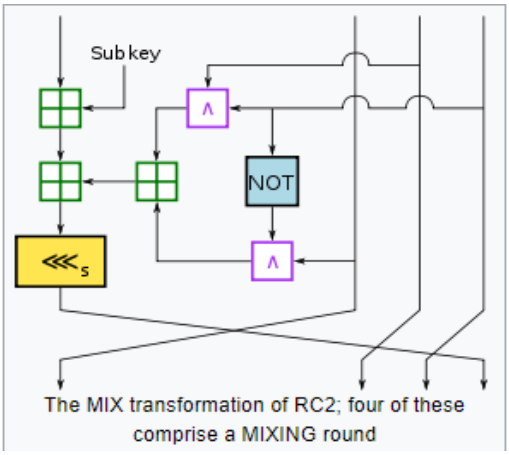
Porovnanie AES a DES

	DES	AES
Vytvorené	1977	1999
Dĺžka kľúča	56 bitov	128, 192 a 256 bitov
Typ šifrovania	Symetrické	Symetrické
Veľkosť bloku	54 bitov	128 bitov
Zabezpečnie	Prokázatelne nedostatočné	Považovaný na bezpečný

Obrázok 9 Porovnanie AES vs DES [8]

RC2

RC2 (tiež známa ako ARC2) je bloková šifra so symetrickým kľúčom, ktorá bola navrhnutá Ronom Rivestom v roku 1987. „RC“ znamená „Ron´s code“ alebo „Rivest cypher“. Ďalšími šiframi, ktoré boli navrhnuté spoločnosťou Rivest sú RC4, RC5 a RC6. RC2 je 64-bitová bloková šifra s kľúčom, ktorý môže mať premenlivú veľkosť. Šifra vykonáva 18 kôl, ktoré sú usporiadané ako zdrojovo ťažká nevyvážená Feistelova sieť, v ktorej 16 kôl jedného typu (mixing) je prerušovaných dvoma kolami druhého typu (masing). [32] Kolo mixing je zložené zo štyroch aplikácií transformácie MIX. RC2 je náchylný k útoku, ktorý súvisí s kľúčom pomocou 234 vybraných obyčajných textov. [9]



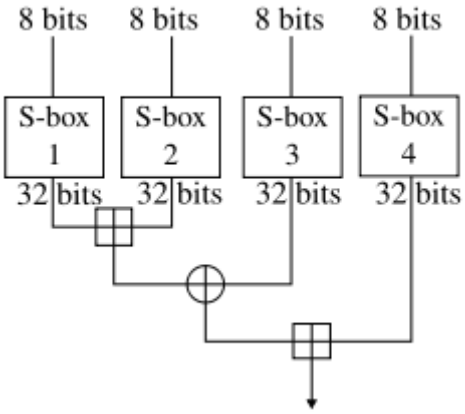
Obrázok 10 Tranformácia MIX v RC2 [9]

Blowfish

Blowfish je symetrická bloková šifra, ktorú navrhol Bruce Schneier v roku 1993 a ktorá bola zverejnená pre verejnosť. Blowfish je zahrnutý vo veľkom počte šifrovaných produktov a šifrovaných balíkov. Jeho bezpečnosť bola overená a dôkladne otestovaná. Ako šifra, ktorá je vo verejnom vlastníctve, bola Blowfish predmetom veľkého množstva kryptoanalýzy a šifrovanie v Blowfish nebolo nikdy prerušené. Taktiež je jedným z najrýchlejších blokových šifier, ktoré je možné získať na verejnosti. Spoločnosť Schneier navrhla Blowfish ako algoritmus, ktorý bude slúžiť na všeobecné použitie, určený ako náhrada za DES. Taktiež je možné ho bez problémov spájať s inými algoritmami. K jeho pozoruhodným vlastnostiam dizajnu patria S-boxy, ktoré sú závislé od kľúčov a tiež veľmi zložitý harmonogram kľúčov.

Blowfish obsahuje 64-bitovú veľkosť bloku a dĺžka jeho kľúča je od 32 bitov do 448 bitov. Je to Feistelova šifra, ktorá vykonáva 16 kôl a používa veľké S-boxy závislé od kľúčov.

Každý riadok predstavuje 32 bitov. Algoritmus uchováva dve podkľúčové polia: 18-vstupné P-pole a štyri 256-vstupné S-boxy. S-boxy prijímajú 8-bitový vstup a produkujú 32-bitový výstup. Jeden záznam P-poľa je používaný každé kolo a po poslednom kole je každá polovica dátového bloku XORedovaná s jedným z dvoch zostávajúcich nepoužitých P-záznamov. [10]



Obrázok 11 Popis Blowfish [10]

F-funkcia rozdelí 32-bitový vstup na štyri osem-bitové štvrťroky a potom použije tieto štvrtiny ako vstup do S-boxov. Výstupy sú pridané modulo 232 a XOR, aby sa vytvoril konečný 32-bitový výstup. Rozvrh kľúčov Blowfish sa začína inicializáciou P-poľa a S-polí

hodnotami, ktoré sú odvodené od hexadecimálnych číslic pí neobsahujúcich žiadny zrejmý vzor. S tajným kľúčom sa vykoná XOR s P-záznamami v poradí. 64-bitový blok sa potom postupne šifruje algoritmom v jeho súčasnej podobe. Výsledný šifrový text nahrádza P1 a P2. Zašifrovaný text sa potom opäť zašifruje novými podkľúčmi a P3 a P4 sa nahradia novým zašifrovaných textom. Takto sa pokračuje, dokým sa nenahradí celé pole P a všetky položky S-boxu. Algoritmus šifrovania bude bežať 521-krát na vygenerovanie všetkých podkľúčov. [10]

Blowfish nepodlieha žiadnym patentom, čo znamená, že je voľne k dispozícii pre kohokoľvek. [10]

Camellia

Camellia je 128 bitová bloková šifra, ktorá bola vyvinutá spoločnosťami Mitsubishi a NTT. Šifra bola schválená na použitie podľa ISO/IEC. Šifra má úrovne zabezpečenia a schopnosti spracovania porovnateľné so šifrou AES(Advanced Encrypthon Standart). Veľkosť bloku šifry Camellia je 16 bajtov (128 bitov) a na šifrovanie sa môžu použiť 128-bit, 192-bit alebo 256-bitové kľúče. Bloková šifra sa navrhla tak, aby ju bolo možné použiť pre softvérové aj hardvérové implementácie, od nízkonákladových kariet Smart až po vysokorýchlostné sieťové systémy. Camellia je šifra, ktorá vykonáva 18 kôl (pri použití 128 bitového kľúča) alebo 24 kôl (pri použití 192 alebo 256 bitového kľúča). Každých šesť kôl sa aplikuje logická transformačná vrstva: takzvaná "FL-function " alebo jej inverzia. Camellia používa štyri 8 x 8-bitové S-boxy so vstupnými a výstupnými transformáciami a logickými operáciami. Šifra tiež používa vstupné a výstupné kľúčové bielenie. Difúzna vrstva používa lineárnu transformáciu založenú na matici MDS. Camellia je patentovaná a k dispozícii v rámci Royalty-Free License. To umožnilo tejto šifre, aby sa stala súčasťou projektu OpenSSL. [11]

OTP

OTP (One-Time Pad) je technika šifrovania, ktorú nemožno prelomiť. Spočíva v tom, že šifrovací kľúč má aspoň takú dĺžku, ako má skutočná správa a pozostáva z celkom náhodných čísel. Každé písmeno obyčajného textu je pridané k jednému prvku z OTP pomocou modulo sčítania. To vedie k šifrovanému textu, ktorý nemá vzťah s pôvodným

textom, keď je kľúč neznámy. Na konci sa ten istý OTP použije na získanie pôvodného textu. Aby to fungovalo, sú povinné tieto pravidlá:

1. OTP by sa mal skladať z celkom náhodných znakov.
2. Kľúč by mal mať rovnakú alebo väčšiu dĺžku ako obyčajný text.
3. Mali by existovať iba dve kópie protokolu OTP.
4. OTP by sa mal používať iba raz.
5. Obe kópie OTP sa zničia okamžite po použití.

Ak sú vyššie uvedené pravidlá prísne dodržané, OTP je úplne bezpečná. Ručné kombinovanie čísel s obyčajným textom je časovo náročná úloha. Preto sa niekedy usudzuje, že OTP už nie je praktické. Moderná počítačová technológia umožňuje celú úlohu kódovania a dekódovania ľahko automatizovať. Napriek tomu sa ručné šifry OTP stále používajú na odosielanie tajných správ agentom prostredníctvom číselných staníc alebo jednosmernýchhlasových spojení (OWVL). [12]

RC4

RC4 (Rivest Cipher 4) je prúdová šifra, ktorá je pozoruhodná svojou jednoduchosťou a rýchlosťou v softvéri. V RC4 sa vyskytlo niekoľko zraniteľností, čo ich robí nezabezpečenými. Zraniteľnosť sa ukazuje hlavne vtedy, keď nie je zahodený začiatok výstupného prúdu kľúčov alebo ak sa používajú súvisiace, či nenulové kľúče. RC4 generuje pseudonáhodný tok bitov. Šifry prúdov sa môžu použiť na šifrovanie kombináciou s obyčajným textom. Dešifrovanie sa vykonáva rovnakým spôsobom. Na vygenerovanie kľúčového prúdu sa používa tajný vnútorný stav, ktorý pozostáva z dvoch častí:

- Permutácia všetkých 256 možných bajtov.
- Dva 8-bitové indexové ukazovatele.

Permutácia sa inicializuje pomocou kľúča, ktorý má premenlivú dĺžku, zvyčajne medzi 40 a 2048 bitmi, s použitím algoritmu plánovania kľúčov (KSA). Po dokončení tohto postupu sa vygeneruje tok bitov prostredníctvom algoritmu pseudonáhodného generovania (PRGA). [13]

Salsa20

Salsa20 je moderná a účinná prúdová symetrická šifra. Je považovaná za dobre navrhnutý a efektívny algoritmus, pri ktorom neexistujú žiadne známe a účinné útoky. Salsa20 pracuje na dátových blokoch s veľkosťou 64 bajtov. Pre každý 64-bajtový dátový blok sa používa funkcia rozšírenia Salsa20, kde vstupom do tejto funkcie je tajný kľúč (ktorého veľkosť môže byť 32 alebo 16 bajtov). Každé volanie funkcie zvyšuje číslo bloku o jeden. Hlavnou podstatou šifry Salsa20 je hašovacia funkcia, ktorá prijíma 64-bajtové vstupné údaje z rozširovacej funkcie Salsa20, zmieša ich a nakoniec vracia 64-bajtový výstup. Hašovacia funkcia Salsa20 pracuje na prijatej postupnosti bajtov, ktorá pozostáva z:

- Tajný klúč
- Nonce s číslom bloku (jedinečné číslo)
- Štyri konštantné vektory prijaté z funkcie rozšírenia, ktorých hodnoty závisia od veľkosti tajného kľúča

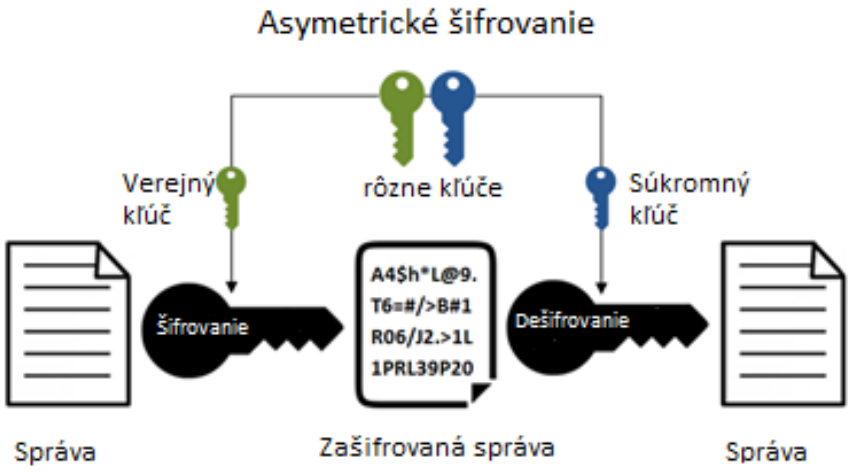
Hašovacia funkcia pracuje s údajmi, ktoré sú rozdelené na slová. Každé slovo obsahuje 4 bajty s hodnotami od 0 do 232. Preto sú vstupné údaje dlhé 16 slov, tajný kľúč obsahuje 8 alebo 4 slová a nonce má 2 slová. Výstup z funkcie rozšírena Salsa20 sa pridá do 64-bajtového bloku údajov XOR. Výsledkom je potom 64-bajtový blok zašifrovaného textu.

Na dešifrovanie zašifrovaného textu sa používa ten istý algoritmus. Údaje sa rozdelia na rovnako veľké časti. Výstup z funkcie rozšírenia Salsa20 sa pridá do 64-bajtového bloku zašifrovaného textu. Výsledkom je 64-bajtový blok pôvodného textu. [14]

1.1.2 Asymetrická kryptografia

Asymetrická kryptografia, ktorá je známa tiež ako kryptografia s verejným kľúčom, používa na zašifrovanie a dešifrovanie textu verejné a súkromné kľúče. Kľúče sú spárované, ale nie sú identické. Verejný kľúč je možné zdieľať so všetkými, zatiaľ čo súkromný kľúč zostane utajený. Na zašifrovanie správy je možné použiť ľubovoľný z týchto kľúčov. Na dešifrovanie sa potom použije opačný kľúč od kľúča, ktorý sa použil na zašifrovanie správy. Súkromný kľúč pozná iba používateľ alebo počítač, ktorý vygeneruje pár kľúčov. Verejný kľúč môže byť známy každému, kto chce poslať zašifrované údaje vlastníkovi súkromného

kľúča. Na základe verejného kľúča sa nedá zistiť súkromný kľúč. Účastníci asymetrického šifrovania sú odosielateľ a prijímateľ. Najprv odosielateľ získa verejný kľúč príjemcu, za pomoci ktorého potom zašifruje správu, čím sa vytvorí zašifrovaný text. Potom sa odošle príjemcovi, ktorý ho dešifruje pomocou svojho súkromného kľúča, na základe ktorého zistí pôvodný text odosielateľa. Jeden odosielateľ však nedokáže prečítať správy iného odosielateľa, aj keď má verejný kľúč príjemcu. Mnoho protokolov ako OpenPGP, S / MIME, SSH a SSL / TLS využíva asymetrickú kryptografiu pre jej šifrovacie funkcie a funkcie digitálneho podpisu. Používa sa aj v rôznych softvérových programoch, ako napríklad prehliadače, v ktorých je potrebné nadviazať bezpečné pripojenie cez nezabezpečenú sieť alebo sa musí overiť digitálny podpis. [15]



Obrázok 12 Asymetrické šifrovanie [16]

Asymetrické šifrovanie sa používa väčšinou v každodenných komunikačných kanáloch, najmä cez internet. Medzi najznámejšie algoritmy využívajúce asymetricky kľúč patrí EIGamal, RSA, DSA, techniky na báze eliptických kriviek. [16]

EIGamal

ElGamal je šifrovací systém, ktorý používa asymetrické šifrovanie kľúčov na komunikáciu medzi dvoma stranami a šifrovanie správy. Tento systém je založený na obtiažnosti nájdenia diskrétneho logaritmu v cyclic group. Aj keď budeme poznať g^a a g^k, je veľmi ťažké vypočítať g^{ak}. [17]

Predpokladajme, že Alice chce komunikovať s Bobom.

1. Bob generuje verejný a súkromný kľúč:
    - Bob si vyberie veľmi vysoké číslo q a cyklic group F_q.
    - Z cyklic group F_q vyberie ľubovoľný prvok g a prvok taký, aby platil výraz gcd (a, q) = 1.
    - Potom vypočíta $h = g^a$.
    - Bob publikuje F, $h = g^a$, q a g použije ako svoj verejný kľúč a zachováva si súkromný kľúč.
2. Alice šifruje údaje pomocou Bobovho verejného kľúča:
    - Alice vyberie prvok k z F tak, že gcd (k, q) = 1.
    - Potom vypočíta $p = g^k$ a $s = h^k = g^{ak}$.
    - Vynásobí premennú s so správou M.
    - Potom odošle $(p, M * s) = (g^k, M * s)$.
3. Bob dešifruje správu:
    - Bob vypočíta $s' = p^a = g^{ak}$.
    - Rozdeľuje M * s pomocou s', aby získal M, kde s = s'.

V tomto kryptosystéme je pôvodná správa M maskovaná vynásobením g^{ak}. Na odstránenie masky sa použije vodítko vo forme g^k. Pokiaľ niekto nevie a, nebude schopný získať M. Je to preto, že nájdenie diskrétneho logaritmu v cyclic group je ťažké a poznanie g^a a g^k nie je postačujúce na výpočet g^{ak}. [17] [34]

RSA Algoritmus

RSA je asymetrický algoritmus, ktorý sa používa na konkrétne bezpečnostné služby alebo účely, umožňuje šifrovanie verejného kľúča a je používaný na zabezpečenie citlivých údajov, hlavne, keď sa odosiela cez nezabezpečenú sieť. V kryptografii RSA môžu verejný aj súkromný kľúč zašifrovať správu. Na jej dešifrovanie sa použije opačný kľúč ako ten, ktorý sa použil na šifrovanie správy. Toto je hlavný atribút, prečo sa RSA stala najpoužívanejším asymetrickým algoritmom. Poskytuje metódu na zabezpečenie dôvernosti, autentickosti, integrity a nevyradenia elektronickej komunikácie a uchovávania údajov. Mnoho protokolov ako Secure Shell, OpenPGP a SSL / TLS sa spolieha na RSA kvôli šifrovaniu a funkciám digitálneho podpisu. Je používaný aj v softvérových programoch

ako prehliadače, ktoré sú jasným príkladom, pretože potrebujú nadviazať bezpečné pripojenie prostredníctvom nezabezpečenej siete (internetu) alebo overiť digitálny podpis. Overenie podpisu RSA patrí medzi najčastejšie vykonávané operácie v sieťových systémoch.

RSA odvodzuje svoju bezpečnosť od zložitosti faktoringu veľkých celých čísel, ktoré sú výsledkom dvoch veľkých prvočísel. Násobenie týchto dvoch čísel je jednoduché, ale určenie pôvodných prvočísel z celkového súčtu alebo faktoringu sa považuje za nerealizovateľné vzhľadom na čas. Algoritmus generovania verejného a súkromného kľúča je najzložitejšou súčasťou kryptografie RSA. Pomocou algoritmu testovania pravosti Rabina-Millera sa vygenerujú dve veľké prvočísla p a q. Modul n sa potom vypočíta vynásobením p a q. Toto číslo používajú verejné aj súkromné kľúče a poskytuje prepojenie medzi nimi. Dĺžka tohto čísla sa nazýva dĺžka kľúča.

Verejný kľúč sa skladá z modulu n a verejného exponentu e, ktorý je nastavený na 65537, pretože je to prvočíslo, ktoré nenadobúda príliš veľké hodnoty. Číslo e nemusí byť tajne vybrané prvočíslo, pretože verejný kľúč je zdieľaný so všetkými.

Súkromný kľúč sa skladá z modulu n a súkromného exponentu d, ktorý sa vypočíta na základe rozšíreného euklidovského algoritmu, ktorý nájde multiplikatívnu inverziu vzhľadom na súčet n. [18]

DSA

Algoritmus digitálneho podpisu (DSA) je federálny štandard, ktorý spracováva informácie pre digitálne podpisy. Je založený na matematickom koncepte modulárnej exponenciácie a problému diskrétneho logaritmu. Algoritmus využíva verejný a súkromný kľúč. Súkromný kľúč sa používa na generovanie digitálneho podpisu pre správu a tento podpis sa môže overiť pomocou zodpovedajúceho verejného kľúča podpisovateľa. Digitálny podpis poskytuje overenie správy (príjemca môže overiť pôvod správy), integritu (príjemca môže overiť, či sa správa od podpisu nezmenila) a nevypovedanie (odosielateľ nemôže tvrdiť, že podpísal správu, pokiaľ ju nepodpísal). [19]

Algoritmus DSA zahŕňa štyri operácie:

- generovanie kľúčov

- distribúcia kľúčov
- podpisovanie
- overovanie podpisov [19]

Z tohto vyplýva že, že DSA má rýchlejší podpis, ale pri kontrole je pomalší. DSA je preto rozumnou voľbou, ak je na strane klienta viac problémov s výkonom. DSA môže byť prevádzkované spoločne s RSA v niektorých serverových systémoch, ako je Apache, čo znamená dodatočnú ochranu. Keďže sú však DSA a RSA podobné, podliehajú podobným útokom. RSA prešla na tvorbu dlhších kľúčov, zatiaľ čo DSA zatiaľ nie. Hoci je vytváranie dlhších kľúčov DSA možné, nerobí sa to. Takže aj napriek tomu, že je DSA rôznymi spôsobmi porovnateľný s RSA, RSA zostáva preferovanou šifrovacou schémou. [20]

Techniky na báze eliptických kriviek (ECC)

Kryptografia eliptických kriviek je technika šifrovania verejných kľúčov, ktorá je založená na teórii eliptických kriviek. Používa sa na vytvorenie rýchlejších, menších a efektívnejších kryptografických kľúčov. ECC negeneruje kľúče pomocou tradičnej metódy generovania, ale generuje ich prostredníctvom vlastností rovnice eliptickej krivky ako produkt veľmi veľkých prvočísel. Táto technológia sa môže používať v kombinácii s väčšinou metód šifrovania (napr. RSA) pomocou verejných kľúčov .

ECC môže poskytnúť úroveň bezpečnosti pomocou 164-bitového kľúča, pričom iné systémy vyžadujú dosiahnutie 1024-bitového kľúča. Keďže ECC pomáha vytvárať ekvivalentnú bezpečnosť s nižšou výpočtovou energiou a využívaním batérie, stáva sa používanou hlavne pre mobilné aplikácie.

Eliptická krivka nie je elipsa, ale je označovaná ako slučková čiara, ktorá pretína dve osi. ECC je založené na vlastnostiach konkrétneho typu rovnice vytvorenej z matematickej skupiny odvodených z bodov, kde priamka pretína osi. Vynásobením bodu na krivke nejakým číslom vznikne nový bod na krivke. Číslo, ktoré bolo použité, je veľmi ťažké nájsť, aj keď je známy pôvodný bod a výsledok.

Rovnice založené na eliptických krivkách majú veľmi užitočnú charakteristiku pre kryptografické účely:

- sú relatívne ľahko uskutočniteľné

- veľmi ťažko sa dajú reverzovať [21]

1.1.3 Hashovacia funkcia

Je to funkcia, ktorá konvertuje dané veľké telefónne číslo na malú celočíselnú hodnotu. Priradená celočíselná hodnota sa potom používa ako index v hash tabuľke.

Dobrá funkcia hash by mala mať nasledujúce vlastnosti:

- efektívne vypočítaná
- rovnomerne distribuované kľúče v hash tabuľke

Napríklad: Pre telefónne čísla je zlá hash funkcia vziať prvé tri číslice. Lepšia funkcia je považovaná vtedy, keď sa vezmú posledné tri číslice.

V praxi sa často využívajú heuristické techniky na vytvorenie dobrej hash funkcie. Počas navrhovania je užitočné poznať kvantitatívne informácie o distribúcii kľúčov. Hash funkcia závisí od každého jedného bitu kľúča, čo znamená, že dva kľúče, ktoré sa líšia jedným bitom alebo jedna skupina bitov nadobúdajú rôzne hash hodnoty. Podobne, ak dva kľúče majú rovnaké bity, ale v inom poradí (napríklad 123 a 213), tiež sa označujú inými hash hodnotami.

Existujú dve heuristické metódy:

1. metóda delenia:
    - vytváranie hash funkcií tak, že sa kľúč priradí do jedného zo slotov tabuľky tým, že zvyšok kľúča (r) sa vydelí veľkosťou tabuľky
    - je rýchla, pretože vyžaduje len jednu operáciu
    - príklad: r = 256 a veľkosť tabuľky = 17, z toho 256/17 = 15 zvyšok 1, takže hash = 1
    - príklad: r = 37596 a veľkosť tabuľky = 17, z toho 37596/17 = 2211 zvyšok 12, takže hash = 12
2. metóda násobenia:
    - vynásobenie kľúča k konštantným skutočným číslom c v rozmedzí $0 < c < 1$ a extrahuje sa zlomková časť z tejto vypočítanej hodnoty k*c

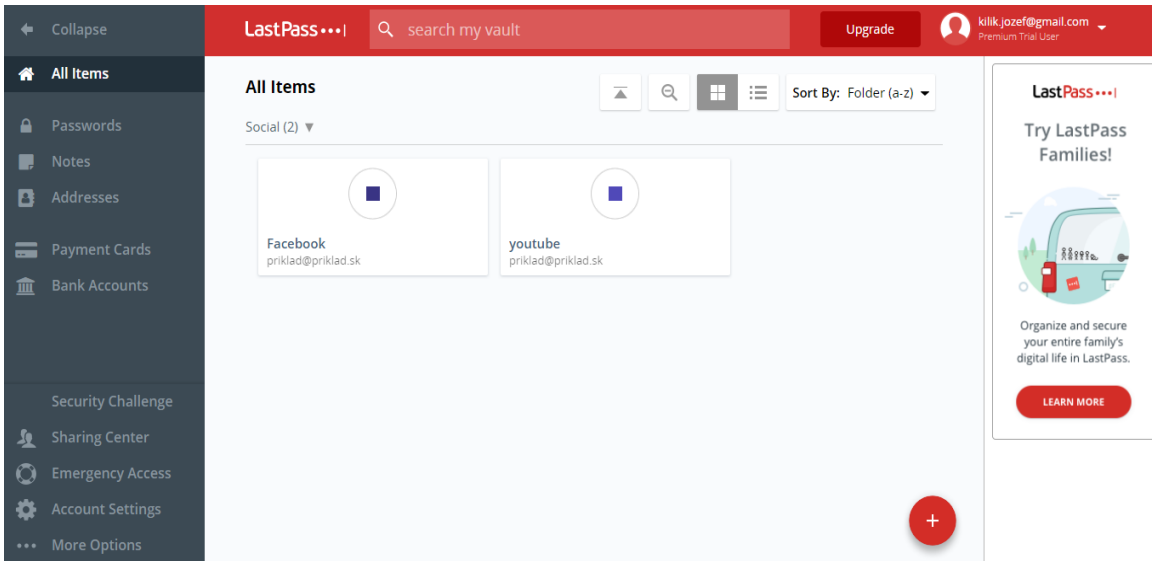
- potom sa vynásobí táto hodnota s veľkosťou tabuľky (m) a vezme sa floor (prirodzené číslo zaokrúhlené nadol) výsledku
- hodnota m nie je kritická a typicky sa vyberie, že je 2 (m = 2p pre niektoré celé číslo p), pretože potom môžeme ľahko implementovať funkciu na väčšine počítačov
- slovo veľkosť stroja je w bitov a kľúč zapadá do jedného slova
- c je zlomok formulára s/(2w), kde s je celé číslo v rozsahu $0 < s < 2W$
- najprv sa musí vynásobiť kľúč s w-bitovým integerom s = c * 2W. Výsledkom bude 2W-bitová hodnota $r_1 * 2w + r_0$, kde r_1 je vyššie poradie slova a r_0 je nižšie poradie slova
- príklad: k = 123456, p = 14, m = 2^14= 16384 a w=32, c sa bude rovnať zlomkovej časti s/2^32, potom k*s = 327706022297664 = (76300 * 2^32) + 17612864. Čiže $r_1 = 76300$ a $r_0 = 17612864$

1.2 Kryptografické softvéry

V tomto digitalizovanom svete je veľmi dôležité, aby boli naše informácie v bezpečí. Či už sa jedná o osobné alebo pracovné vzťahy, vždy nám hrozí, že sa dostanú do nesprávnych rúk. Jednou z najlepších a najbezpečnejších dostupných metód ochrany digitálnych údajov je ,ako sme si už spomínali, šifrovanie. Šifrovanie slúži na zabezpečenie citlivých informácií pred počítačovými zlodejmi alebo inými online hrozbami. Tiež je to spôsob, akým sa dá archivovať veľké množstvo údajov alebo zabezpečiť súkromná komunikácia cez internet. Internetoví používatelia, ako aj organizácie môžu využiť bezplatné šifrovacie nástroje, vďaka ktorým sa zvýši ochrana tým, že uchovávajú cenné informácie pred nebezpečnými účastníkmi. Šifrovanie citlivých údajov je nevyhnutné bez ohľadu na to, či sú uložené na počítači, lokálne alebo odoslané cez internet. Existuje mnoho šifrovacích nástrojov, ktoré slúžia na zaistenie bezpečnosti a zabezpečenia najcennejších údajov, no podľa spoločnosti HeimdalSecurity sa medzi najlepšie nástroje zaraďujú: [23]

1. LastPass

- Jeden z najobľúbenejších softvérov na správu hesiel.
- Používa sa zadarmo s obmedzenými funkciami, ale stále bude zabezpečovať heslá a osobné údaje svojich používateľov.
- Pri použití tohto šifrovacieho softvérového nástroja už nie je potrebné pamätať si ani zapisovať heslo, pretože bude uložené v tomto nástroji.
- Ľahko použiteľné a intuitívne rozhranie, ktoré zjednodušuje život používateľov.
- Existujú rozšírenia pre webové prehliadače, napr. Mozilla Firefox a Google Chrome. Dá sa použiť aj mobilná aplikáciu, ktorá je k dispozícii pre Android a Apple.
- Odporúča používateľom nastaviť jedinečné a ťažko prelomiteľné heslo a tiež ich odrádza, aby ho používali na viacerých miestach. V prípade ak používa rovnaké heslo pre viac účtov, nástroj odporučí zvoliť iné heslo. [23]



Obrázok 13 Prostredie LastPass [vlastné]

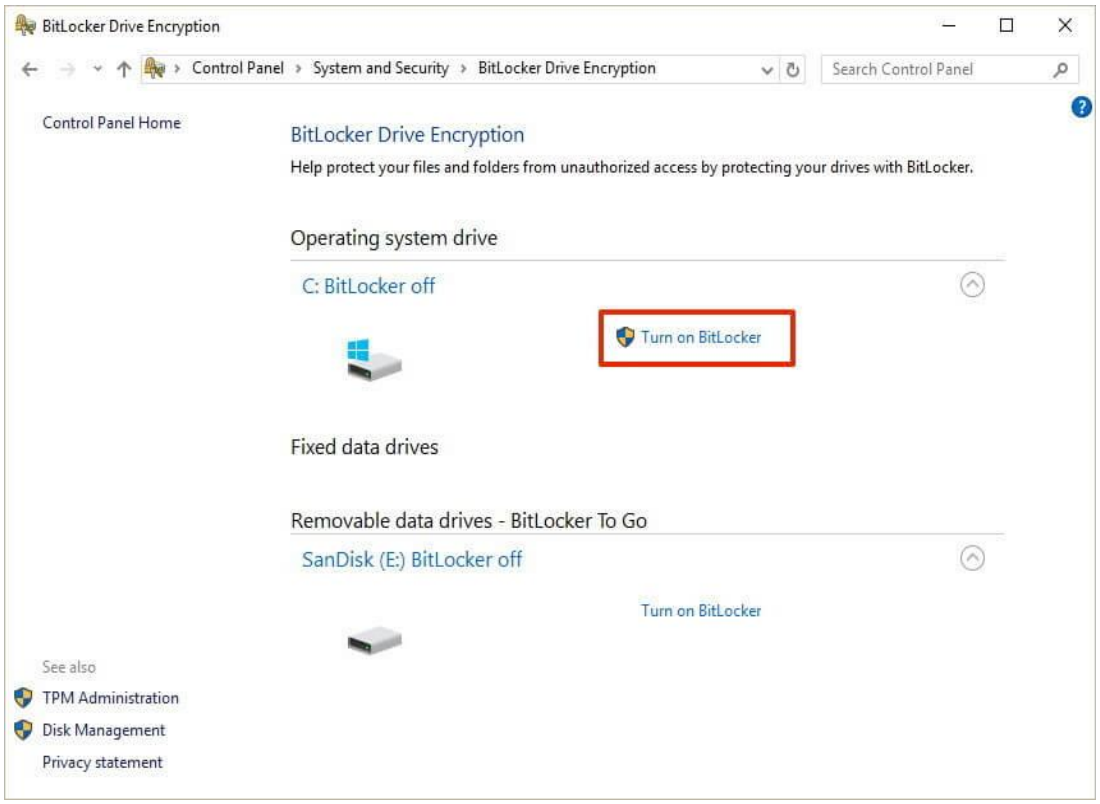
2. BitLocker

- Je softvér spoločnosti Microsoft, ktorý sa používa na šifrovanie konkrétnej časti disku alebo celého pevného disku.

- Je nástroj na šifrovanie disku zabudovaný v operačných systémoch Windows 10, ktorý používa symetrickú šifru AES (128 a 256 bitov) na šifrovanie údajov.
- AES bol testovaný a vylepšený a momentálne ho používa väčšina dodávateľov zabezpečenia kvôli jeho vysokej úrovni bezpečnosti a optimalizácie.

Výhody, prečo by si mal používateľ zvoliť nástroj BitLocker:

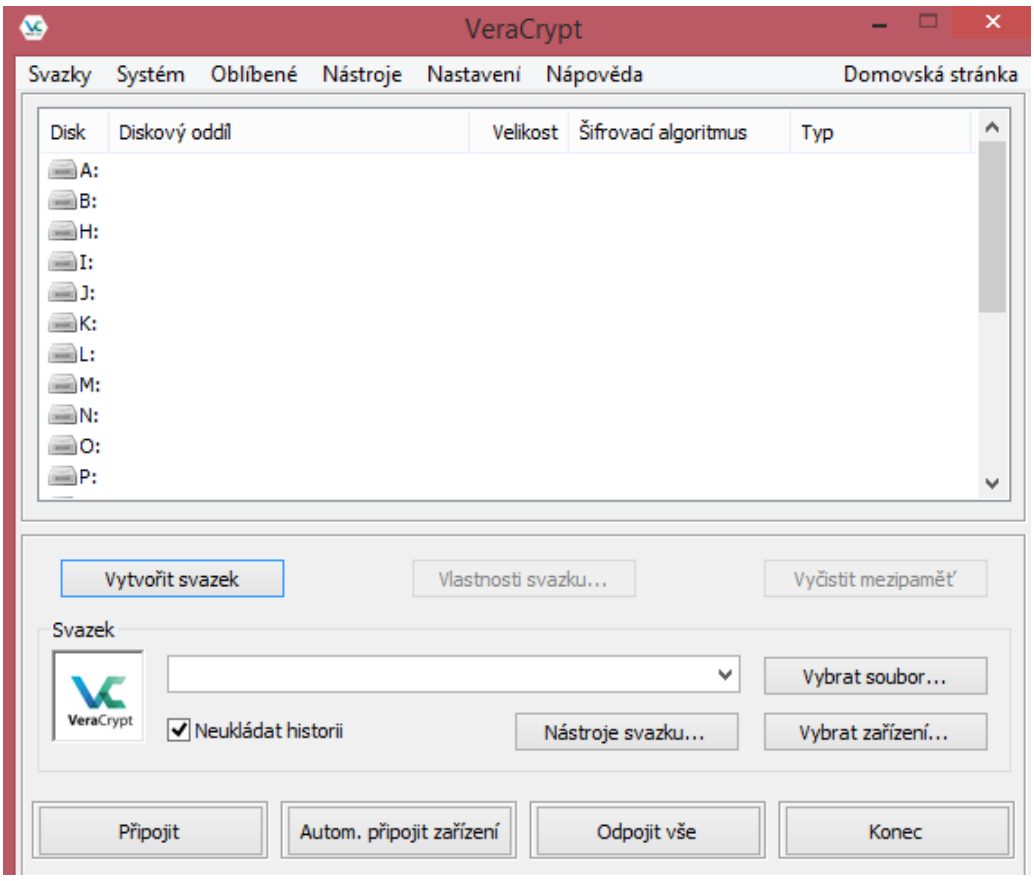
- Je ľahko používateľný a je integrovaný do operačného systému Windows, takže nie je potrebné pridávať ďalší šifrovací softvér.
- Ide o bezplatný softvér, ktorý sa používa na šifrovanie údajov. Slúži na zabránenie narušeniam údajov a ukradnutiu údajov z pevného disku.
- Šifruje celý disk, čo znemožňuje zločincom ukradnúť laptop, odstrániť v ňom pevný disk a získať prístup k súborom používateľa.
- Pokiaľ je aktivované šifrovanie pomocou BitLocker a pridajú sa nové súbory, nástroj BitLocker ich automaticky zašifruje. [23]



Obrázok 14 Prostredie BItLocker [23]

3. VeraCrypt

- Je bezplatný šifrovací softvérový nástroj, ktorý je možné použiť na operačných systémoch Windows, Mac OS X a Linux.
- Podobne ako BitLocker, tiež podporuje Advanced Encryption Standard (AES) a umožňuje skryť šifrované zväzky v iných zväzkoch.
- Je to program ktorý má otvorený zdrojový kód, čo znamená, že ktokoľvek môže zdrojový kód stiahnuť a používať.
- Tento šifrovací softvér je výbornou alternatívou k nástroju TrueCrypt a stále zlepšuje a vylepšuje zabezpečenie. [23]
- Vytvára virtuálny šifrovaný disk v súbore a pripojí ho ako skutočný disk.
- Šifruje celý oddiel alebo úložné zariadenie, napríklad jednotku USB flash alebo pevný disk.
- Šifrovanie je automatické, v reálnom čase a transparentné. [24]



Obrázok 15 Prostredie VeraCrypt [vlastné]

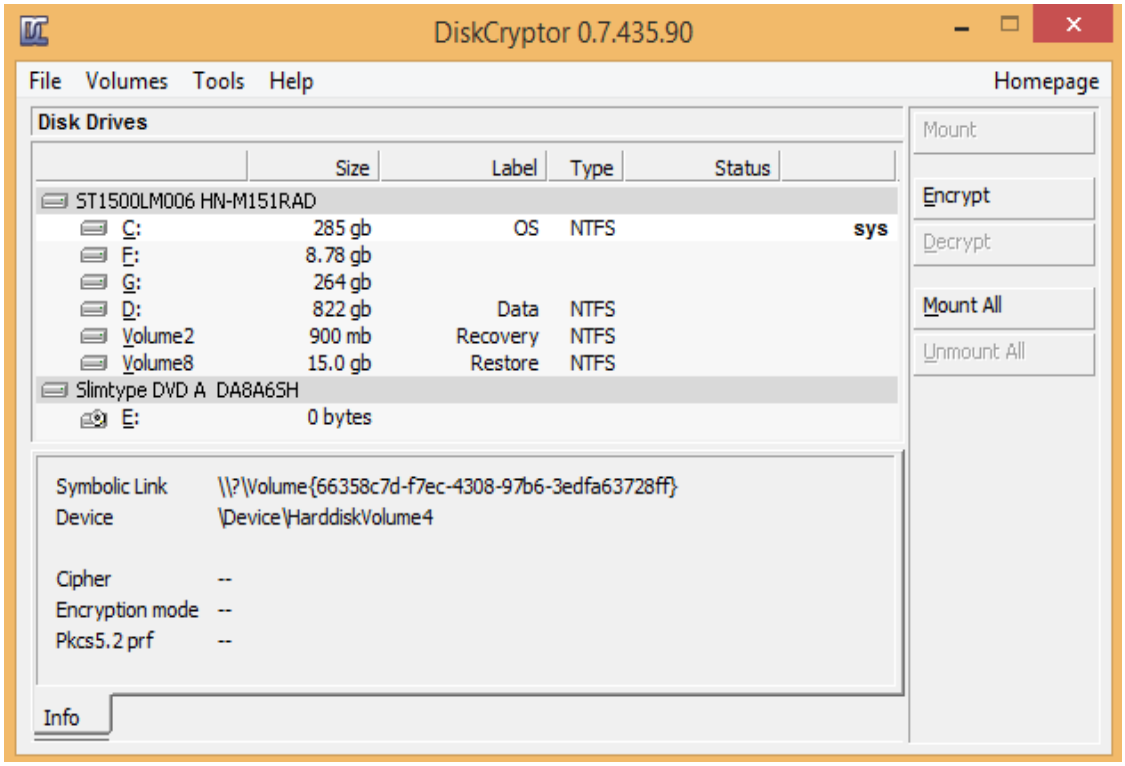
4. FileVault 2

- Umožňuje šifrovať údaje, ktoré sú uložené na zariadeniach s Mac OS a Mac hardvérom.
- Je bezplatný šifrovací softvérový nástroj na šifrovanie citlivých údajov používateľa.
- Podobne ako nástroje BitLocker a VeraCrypt, FileVault 2 používa šifrovanie XTS-AES-128 s 256-bitovým kľúčom. [23]
- Je celo-diskový šifrovací program, ktorý šifruje údaje v počítači Mac na zabránenie neoprávnenému prístupu od kohokoľvek, kto nevlastní dešifrovací kľúč.
- Podporuje hardvér aj pre zariadenia, ktoré už spoločnosť Apple nepodporuje.
- Môže byť spravovaný lokálne alebo centrálne, buď používateľmi alebo IT oddelením.
- Šifrovanie celého disku chráni všetky údaje, ktoré sú uložené na disku teraz aj v budúcnosti.
- Ak je potrebné zmeniť prístupové heslo alebo obnovovací kľúč, musí sa dešifrovať celý zväzok a šifrovací proces sa musí znova spustiť s novým kľúčom. [25]

Obrázok 16 Prostredie FileVault [23]

5. DiskCryptor

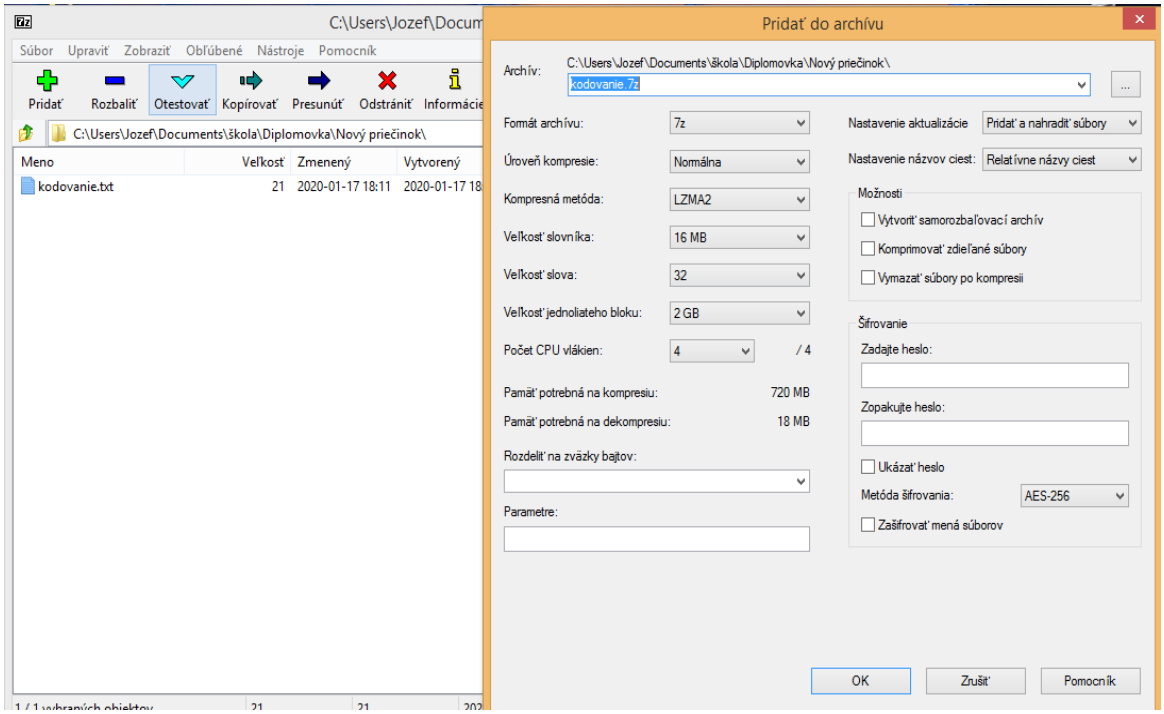
- Je otvorený a bezplatný šifrovací softvér, ktorý sa používa na zabezpečenie internej a externej jednotky, systémového oddielu a obrázkov ISO, jednotiek USB flash alebo rôznych iných úložných zariadení.
- Rovnako ako softvér BitLocker, predstavuje šifrovací nástroj celého disku pre operačný systém Windows a môže využívať viac šifrovacích algoritmov, ako sú AES, Serpent, Twofish.
- Rozhranie je jednoduché a intuitívne. Stačí vybrať, čo sa má zašifrovať a kliknutím na „Šifrovať“ sú tieto údaje zabezpečené. [23]
- Plná kompatibilita so zavádzačmi tretích strán (LILO, GRUB).
- Šifrovanie systémových a bootovacích oddielov je pred zavedením zabezpečené autentifikáciou.
- Je umožnené umiestnenie zavádzača na externé médium a tiež autentifikáciu pomocou kľúčového média. [26]
- Plná podpora pre šifrovanie externých zariadení USB.



Obrázok 17 Prostredie DiskCryptor [vlastné]

6. 7-ZIP

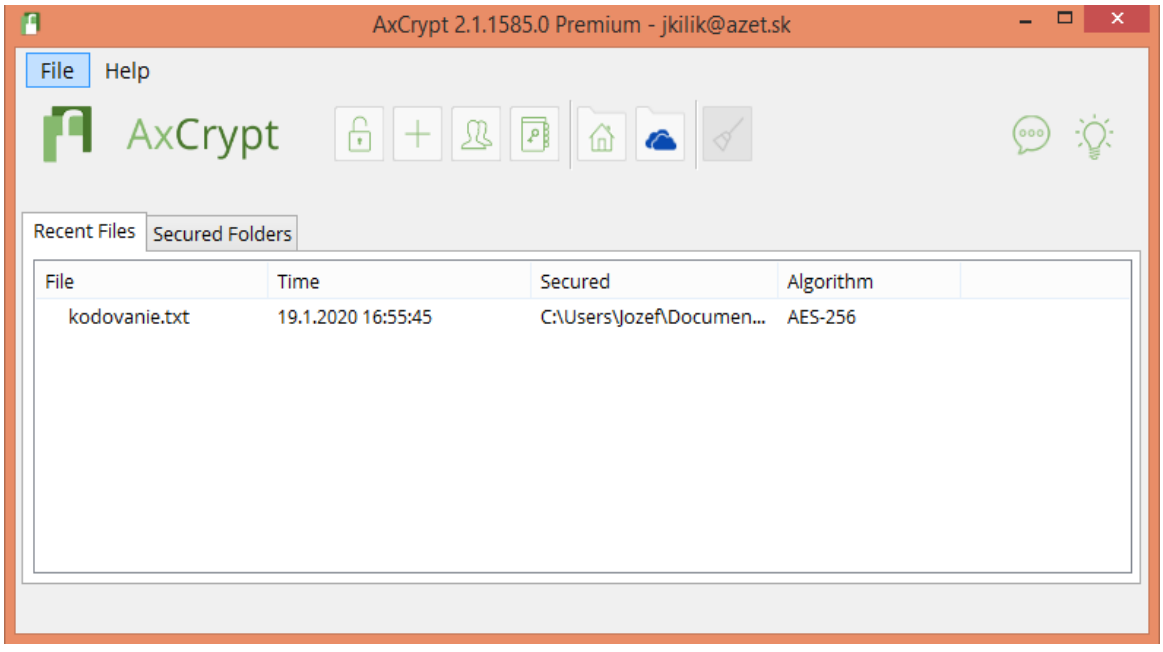
- Je šifrovací softvér s otvoreným zdrojovým kódom a je známy svojou jednoduchosťou.
- Dokáže extrahovať väčšinu archívov a používa silné šifrovanie AES-256.
- Nešifruje celý pevný disk, ale iba konkrétne súbory a dokumenty, v ktorých sa nachádzajú citlivé informácie. [23]
- Dodáva sa s nástrojom File manager spolu so štandardnými nástrojmi archivátora.
- Zväzky dynamicky premenlivých veľkostí, ktoré umožňujú použitie na zálohovanie na vymeniteľných médiách (zapisovateľné disky CD a DVD)
- Umožňuje rozbaľovanie archívov s poškodenými názvami súborov a podľa potreby ich premenuje. [27]



Obrázok 18 Prostredie 7-zip [vlastné]

7. AxCrypt

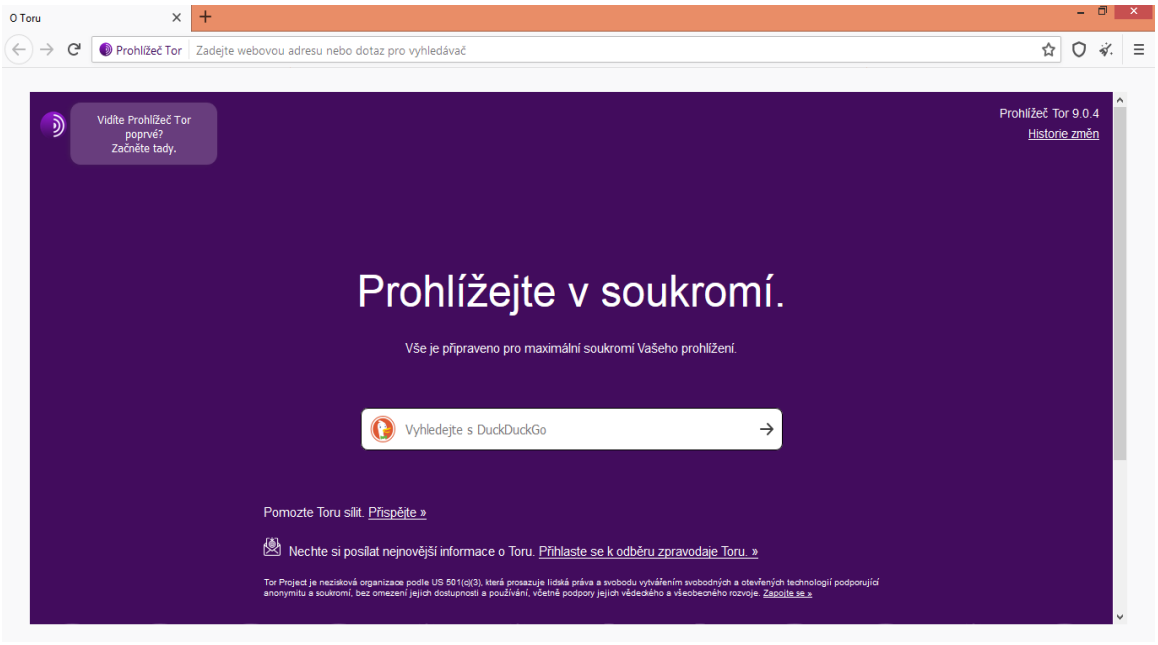
- Je šifrovací nástroj, ktorý má podobne ako 7-ZIP otvorený zdrojový kód, ktorý môže ktokoľvek použiť.
- Ponúka bezplatné riešenie a verziu pre systémy Windows, Android, Mac OS a iOS.
- Je vybavený šifrovacím algoritmom AES-256 a dokáže efektívne šifrovať súbor, celú zložku alebo skupinu súborov.
- Súbory sa môžu šifrovať na určité časové obdobie alebo sa automaticky dešifrujú, keď tento súbor dosiahne cieľ.
- Automaticky zabezpečené súbory v online úložiskách Dropbox alebo Google disk.
- Umožňuje otváranie zašifrovaných súborov ostatnými používateľmi pomocou ich vlastného hesla.
- Spravovanie svojich hesiel a bezproblémový prístup ku nim z ktoréhokoľvek miesta. [28]



Obrázok 19 Prostredie AxCrypt [vlastné]

8. Tor Browser

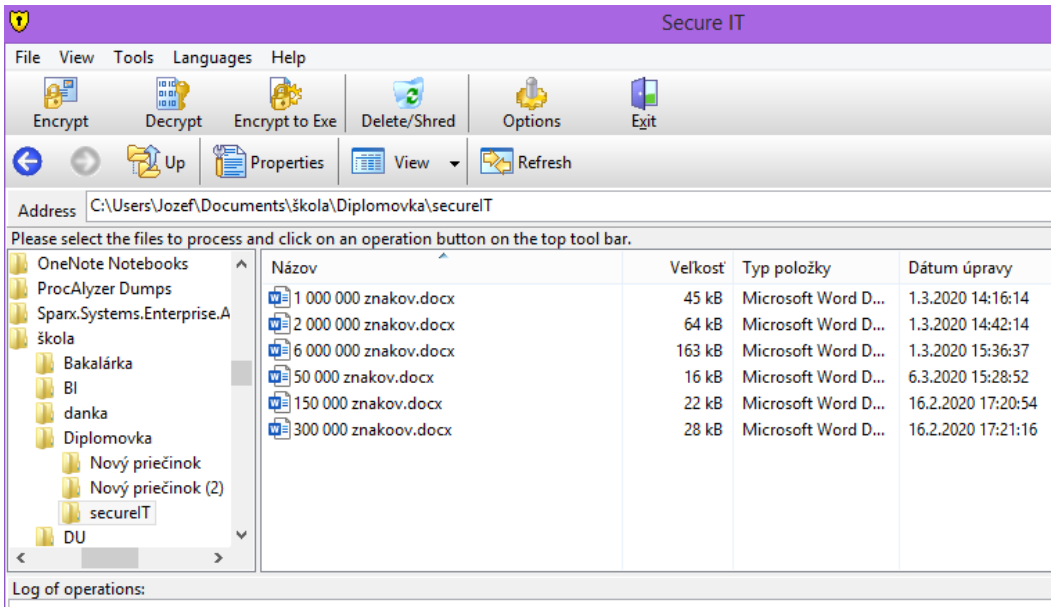
- Je prehliadač anonymného prehliadania internetu, ktorý je vynikajúcou možnosťou na šifrovanie online prevádzky používateľa a zabránenie nezvaným hosťom pri prehliadaní jeho aktivít.
- Blokuje populárne doplnky prehliadača, ako sú RealPlayer, Flash, Quicktime a ďalšie, s ktorými sa pri zistení IP adresy používateľa dá manipulovať.
- Neodporúča sa inštalovať doplnky, pretože môžu obísť Tor a ohroziť súkromie a bezpečnosť používateľa.
- Bol navrhnutý pre každého používateľa, ktorý chce zakryť akúkoľvek aktivitu prehliadania pred niekým iným.
- Používanie šifrovacieho softvéru, ako je Tor, sťaží sledovanie vašich online aktivít zločincami. [35]



Obrázok 20 Prostredie Tor [vlastné]

9. Secure IT 2000

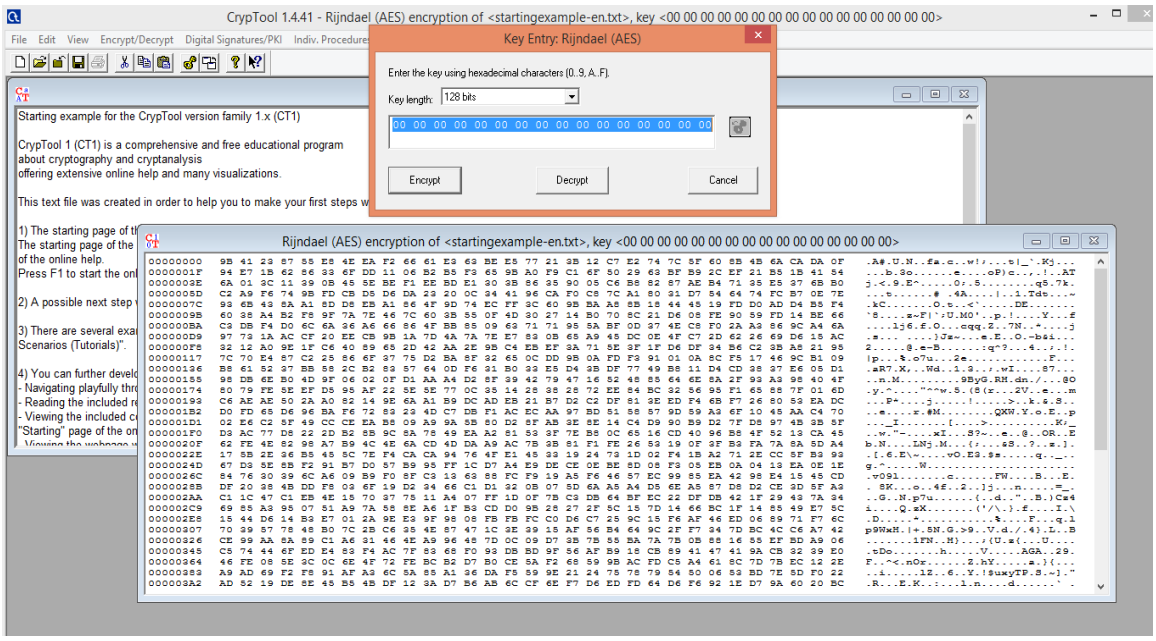
- Je to šifrovací program, ktorý slúži na šifrovanie jednotlivých súborov.
- Umožňuje komprimovanie súborov, čo znamená, že dĺžka trvania tohto procesu je o niečo dlhšia ako u niektorých iných programoch, no súbory sú potom lepšie spravovateľné. [31]
- Vyžaduje heslo na prístup ku ktorémukoľvek zašifrovanému súboru, toto heslo si je potrebné uchovať v bezpečí. V prípade, že sa heslo stratí, nebude umožnený prístup k žiadnym zo svojich šifrovaných súborov a Secure IT nemá proces obnovenia hesla.
- Obsahuje možnosť skartovania súborov, ktorá pomáha vymazať všetky dáta z pevného disku.
- Nie je náročný na ovládateľnosť, takže je ľahké zistiť, ako šifrovať súbory.
- Umožňuje výber medzi dvoma rôznymi šifrovacími algoritmami AES alebo BLOWFISH.
- Má podobný vzhľad ako prehľadávač súborov Windows, kde je možné vybrať jednotlivé dokumenty alebo celú zložku, ktorá sa zašifruje. Taktiež umožňuje pravým kliknutím myši na súbor na pracovnej ploche zobraziť možnosť šifrovania. [31]



Obrázok 21 Prostredie secure IT [vlastné]

10. Cryptool 1 (CT1)

- Je bezplatný program na kryptografiu a kryptanalýzu pre systémy Windows.
- Umožňuje prácu v 6 jazykoch a je to najrozšírenejší e-learningový softvér svojho druhu.
- Bol navrhnutý ako interná podniková aplikácia, ktorá slúžila na školenie v oblasti informačnej bezpečnosti, no odvtedy sa vyvinul v dôležitý open-source program v oblasti kryptológie a bezpečnosti IT.
- Obsahuje klasické a moderné kryptografické algoritmy (šifrovanie a dešifrovanie generovanie kľúčov atď.).
- Umožňuje vizualizáciu niektorých algoritmov (RSA, Diffie-Hellman, digitálne podpisy, AES, atď.).
- Vykonáva kryptoanalýzu pre niekoľko algoritmov (RSA, AES, Vigenère, atď.).
- Umožňuje vykonávať metódy kryptoanalytického merania (autokorelácia, entropia, n-gramy, atď.).



Obrázok 22 Prostredie Cryptoolu [vlastné]

Mobilné aplikácie

Medzi šifrovacie nástroje sa zaraďujú aj mobilné aplikácie na šifrovanie správ. Bez úplného šifrovania (end-to-end) sa konverzácie môžu dostať do rúk kybernetických zločincov alebo iných škodlivých činiteľov. Úplné šifrovanie (end-to-end) znamená šifrovanie komunikácie tak, aby sa k nim nevedela dostať tretia strana. To znamená, že keď prebieha komunikácia medzi dvomi alebo viac zariadeniami prostredníctvom aplikácie, ktorá obsahuje toto šifrovanie, informácie sa budú prenášať pomocou tajného kódu a nie nezabezpečeného textu. Podľa spoločnosti HeimdalSecurity medzi najviac zabezpečené aplikácie na šifrovanie správ sú:

- WhatsApp – v roku 2016 implementoval end-to-end šifrovanie, takže používatelia môžu komunikovať bezpečnejšie a ak by počítačoví zločinci porušili WhatsApp, nemohli by konverzácie používateľov dešifrovať, pretože ich neukladajú na svoje servery. Aplikácia sa dá využívať zadarmo.
- Viber - od roku 2017 ponúka táto aplikácia veľa užitočných funkcií, ktoré používateľom zvýšili zážitky z telefonovania a všetky z nich sú zabezpečené s end-to-end šifrovacím systémom. Aplikácia sa dá stiahnuť zadarmo a používa ju viac ako 900 miliónov používateľov.
- Threema – v aplikácii sa po doručení odoslané správy odstránia zo svojich serverov. Okrem toho má funkciu s názvom „Súkromné čety“, ktorá pomáha používateľom chrániť svoje rozhovory pomocou PIN kódu. Týmto spôsobom sú dôverné rozhovory chránené pred tretími stranami. Túto aplikácia je nutné si kúpiť za 2,99€.

Ďalšie mobilné šifrovacie aplikácie sú napríklad Line, Telegram, KakaoTalk a Dust. [30]

2 Cieľ a metodika práce

Táto kapitola bude obsahovať cieľ diplomovej práce a taktiež bude popisovať metodiku podľa ktorej sa vypracovávala práca.

2.2 Cieľ a metodika práce

Cieľom tejto práce je vybrať rôzne prípadové úlohy, ktoré sa budú šifrovať viacerými šifrovacími algoritmami za pomoci rôznych šifrovacích softvérov a na základe toho porovnať výsledky a efektivitu šifrovania. Aby bolo možné tento cieľ dosiahnuť, bude potrebné rozpracovať čiastkové ciele.

Prvým čiastkovým cieľom je bližšie sa zoznámiť s kryptografiou a nájsť jednotlivé šifrovacie algoritmy, ktoré budú vhodné na použitie pre prácu. V dnešnej dobe existuje mnoho takýchto algoritmov, takže je veľmi dôležité vybrať také, ktoré budú vhodné pre šifrovanie prípadových úloh.

Druhým čiastkovým cieľom je výber a popis vhodných šifrovacích softvérov na šifrovanie. Existuje veľa druhov šifrovacích softvérov, ktoré šifrujú nielen obyčajný text, ale taktiež aj súbory, priečinky alebo celé disky. Základom pre vypracovanie týchto dvoch čiastkových cieľov je získaná a preštudovaná literatúra vyznačená na konci záverečnej práce. Použili sa prevažne odborné publikácie a vedecké články, ktoré sú dostupné na internete. Pomocou analýzy a pozorovania sa určia najvhodnešie šifrovacie algoritmy a najvhodnejšie šifrovacie softvéry.

Po splnení týchto čiastkových cieľov, čiže získaní znalostí o vhodných softvéroch a o vhodných algoritmoch, sa prípadové úlohy začnú šifrovať podľa vybratých šifrovacích algoritmov vo vybratom šifrovacom softvéri a zo získaných výsledkov porovnávať ich jednotlivé atribúty. Táto kapitola je založená na získaní znalostí, na základe ktorých môžeme dokončiť celkový cieľ a vyhodnotiť ho.

3 Výsledky práce a diskusia

V tejto kapitole sa nachádza postupné vypracovávanie cieľa práce. Tento cieľ je výber rôznych prípadových úloh, ktoré sa zašifrujú viacerými šifrovacími algoritmami za pomoci rôznych šifrovacích softvérov. Následne sa budú porovnávať jednotlivé atribúty šifrovania týchto úloh.

3.1 Výsledky a diskusia

Pre naplnenie cieľa sme si vybrali 3 prípadové úlohy, kde jedna je text s 50 000 znakmi, druhá je text s 150 000 znakmi a tretia je text so 300 000 znakmi. Tieto tri prípadové úlohy budeme šifrovať pomocou symetrického šifrovacieho algoritmu AES a pomocou asymetrického šifrovacieho algoritmu RSA. Tieto šifrovania budeme vykonávať v rôznych šifrovacích softvéroch. Najčastejšie budeme používať softvér Cryptool, no na porovnanie využijeme aj softvéry ako AxCrypt, 7-Zip File Manager a Secure IT. V teoretickej časti sme si bližšie priblížili tieto šifrovacie algoritmy a šifrovacie softvéry, s ktorými budeme pracovať.

Najprv začneme šifrovať naše prípadové úlohy algoritmom AES v rôznych softvéroch, kde v každom softvéri budeme popisovať jednotlivé kroky, ako je možné našu prípadovú úlohu zašifrovať a potom opätovne odšifrovať. Zašifrujeme všetky tri prípadové úlohy a následne odprezentujeme výsledné hodnoty. Tie si potom uchováme na ďalšie porovnávania s výsledkami od ostatných softvérov. Na základe týchto výsledkov sa určí, ktorý softvér je najefektívnejší na šifrovanie prostredníctvom algoritmu AES.

Takto postupne budeme šifrovať prípadové úlohy aj ďalším algoritmom RSA. Každé jedno šifrovanie bude mať svoje porovnávanie a vyhodnocovanie vzhľadom na algoritmus. Po šifrovaní sa zostaví záverečné vyhodnotenie, ktoré určí celkovú efektivitu jednotlivých algoritmov v softvéroch. A na záver sa vykoná porovnanie medzi šifrovaním súborov týmito algoritmami v softvéri Cryptool.

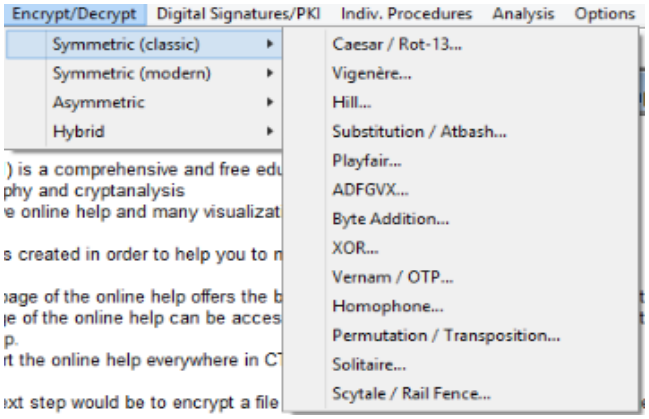
3.1.1 Použitie šifrovacieho algoritmu AES

Ako sme sa už dozvedeli z teoretickej časti, AES patrí medzí symetrické algoritmy a používa jeden kľúč na šifrovanie a dešifrovanie dát. Dĺžka tohto kľúča sa skladá z 128, 196 alebo 256 bitov. Je to blokový šifrovací algoritmus, ktorý sa aplikuje na dáta, kde má pevne stanovenú dĺžku - najčastejšie 128 bitov. Keď sú dáta, ktoré sa šifrujú dlhšie, ich spracovávanie sa vykonáva po jednotlivých blokoch. Ak sú však šifrované dáta kratšie, je potrebné ich doplniť, aby dosahovali stanovenú dĺžku.

Cryptool

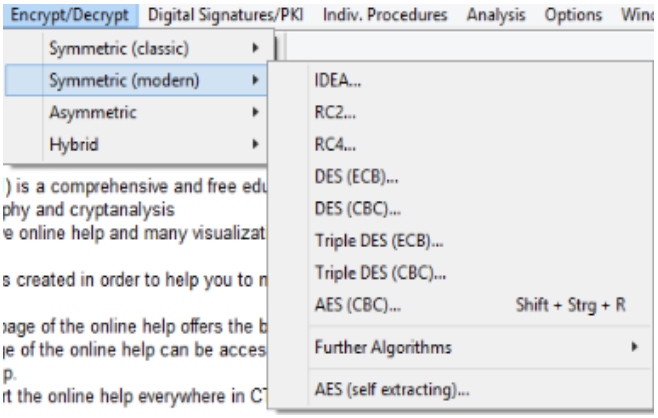
Prvým softvérom, ktorým budeme šifrovať prípadové úlohy, je **Cryptool**. Na začiatku si musíme najprv vybrať, v ktorom algoritme chceme šifrovať. Cryptool ponúka možnosti šifrovania v obrovskom množstve algoritmov. Ponúka 4 druhy algoritmov:

- Symetrické jednoduché



Obrázok 23 Symetrické jednoduché (vlastné)

- Symetrické moderné

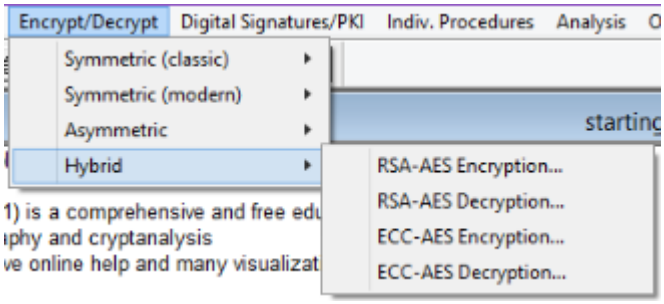


Obrázok 24 Symetrické moderné (vlastné)

- Asymetrické

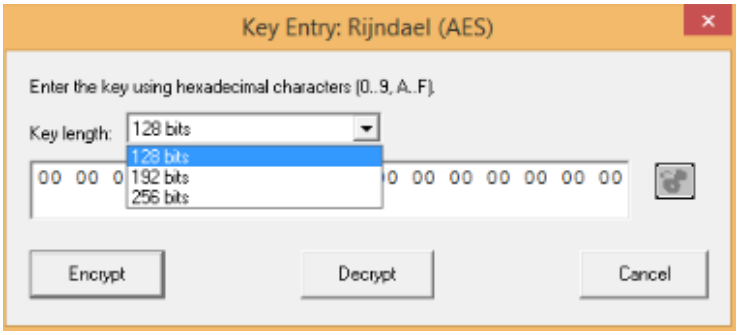
Obrázok 25 Asymetrické (vlastné)

- Hybridné



Obrázok 26 Hybridné (vlastné)

Použijeme teda algoritmus AES, ktorý vyberieme zo symetrických moderných šifrovacích algoritmov. Pri výbere tohto algoritmu musíme určiť pevne stanovenú veľkosť kľúča. Túto veľkosť nastavíme na 128 bitov.



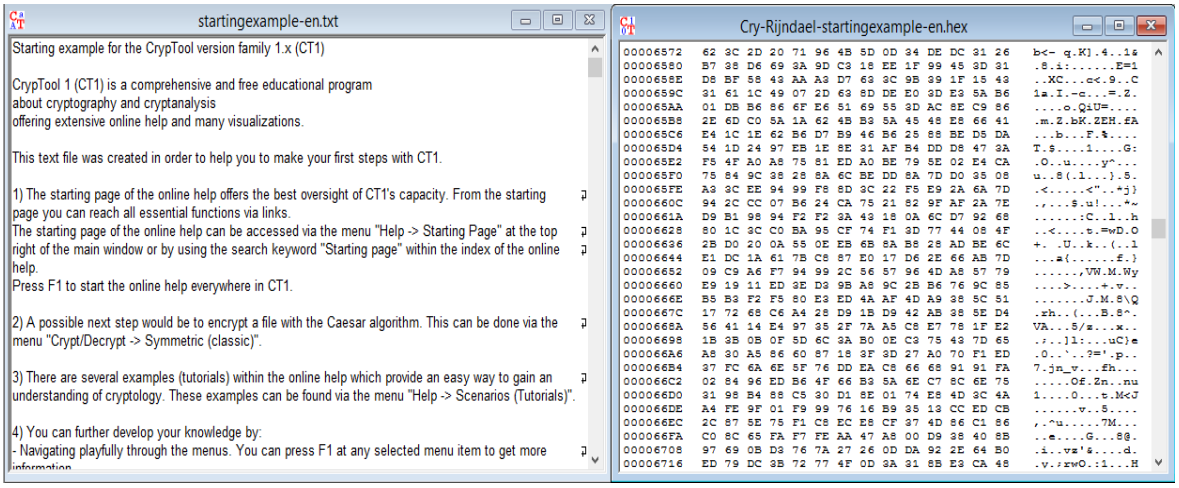
Obrázok 27 Výber veľkosti kľúča (vlastné)

Po výbere algoritmu vyplníme textové pole prvou prípadovou úlohou s 50 000 počtom znakov.

Výsledok prvej prípadovej úlohy prostredníctvom algoritmu AES softvérom CrypTool:

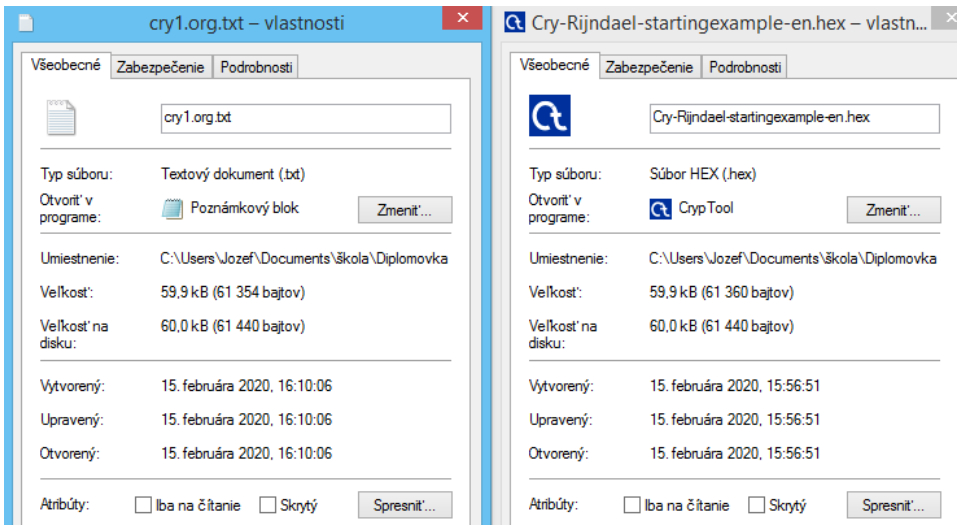
Pri zadaní všetkých potrebných náležitostí pre zašifrovanie prvej prípadovej úlohy s 50 000 znakmi nám vznikol nový zašifrovaný text. Tento text je zložený zo 122 720 znakov, čo je takmer 3 krát viac ako pôvodný text. Rýchlosť, ktorou CrypTool zašifroval tento text

bola extrémne rýchla, keďže hneď po kliknutí na tlačítko „Encrypt“ sa objavil nový zašifrovaný text. Toto znamená, že čas odozvy je takmer okamžitý a podľa rozsahu znakov sa zdá, že je nemožné prebiť túto šifru. Dešifrovanie šifrovaného textu prebieha podobne ako šifrovanie. Po stlačení tlačidla „decrypt“ sa znovu ihneď zobrazí pôvodný text. Na obrázku nižšie je znázornený pôvodný text s názvom „startingexample-en.txt“ a šifrovaný text s názvom „Cry-Rijndael-startingexample-en.hex“, ktorý sme dostali po šifrovaní.



Obrázok 28 Pôvodný a šifrovaný text (vlastné)

Veľkosť súboru pôvodného textu je 61 354 bajtov a veľkosť nového zašifrovaného súboru je 61 360 bajtov. Je to takmer totožná hodnota a na disku aj obsadzuje rovnakú hodnotu 60 kB.

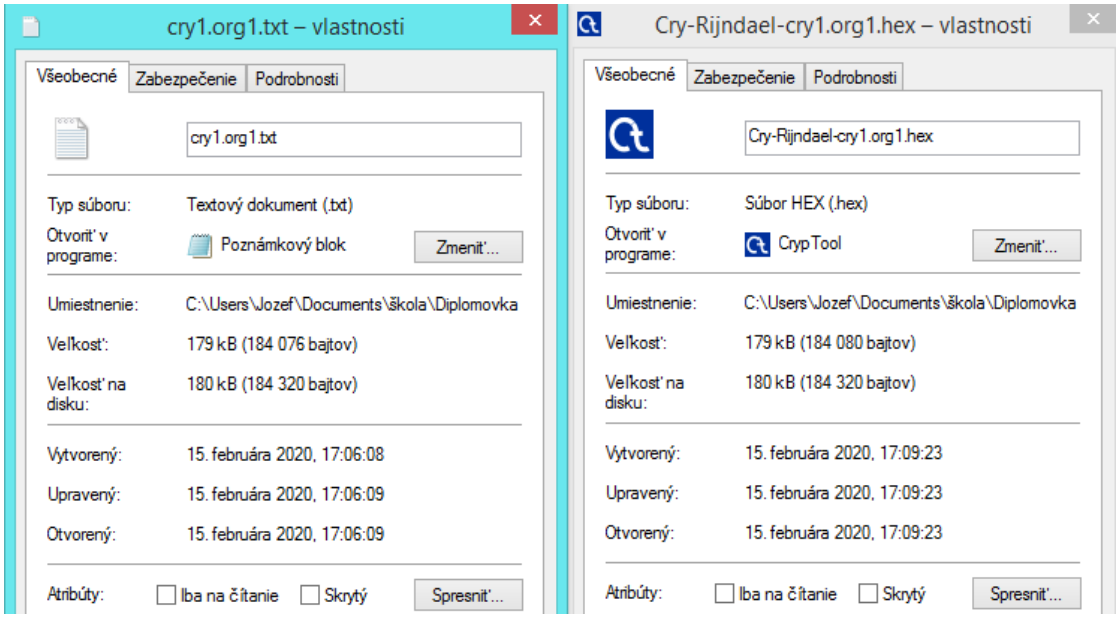


Obrázok 29 Vlastnosti súborov prvej úlohy (vlastné)

Po získaní týchto výsledkov sme zistili, že prvá prípadová úloha sa vykonala, čo sa týka rýchlosti, takmer okamžite. Veľkosť súboru sa síce zväčšila z 50 000 znakov na 122 720 znakov, no v konečnom dôsledku jeho veľkosť na disku sa vôbec nezmenila. Môžeme teda povedať, že táto prípadová úloha pomocou šifrovacieho algoritmu AES a softvéru CrypTool je efektívna.

Výsledok druhej prípadovej úlohy prostredníctvom algoritmu AES softvérom CrypTool:

Druhou prípadovou úlohou je text so 150 000 znakmi. Po vložení tohto textu do softvéru a stlačenia tlačítka „encrypt“ sa znovu vykonalo toto šifrovanie ihneď. To znamená že odozva času je takmer okamžitá. Zašifrovaný text sa skladá zo 368 160 znakov, čo znamená, že počet znakov sa zvýšil až o 218 160 znakov. Keďže ide o rovnakú šifru, tak zašifrovaný text je podobný ako je na obrázku v prvej prípadovej úlohe. Dešifrovanie prebehlo podobne ihneď po kliknutí na tlačidlo „decrypt“ s minimálnym časom odozvy. Veľkosť súboru pôvodného textu je 184 076 bajtov, čo je zhruba trikrát viac ako veľkosť pôvodného textu prvej prípadovej úlohy. Veľkosť šifrovaného textu je znovu zanedbateľne odlišná, pretože pozostáva z 184 080 bajtov, čo je iba o 4 bajty viac ako pôvodný text. Znova teda platí, že na disku majú rovnakú veľkosť 180 kB.

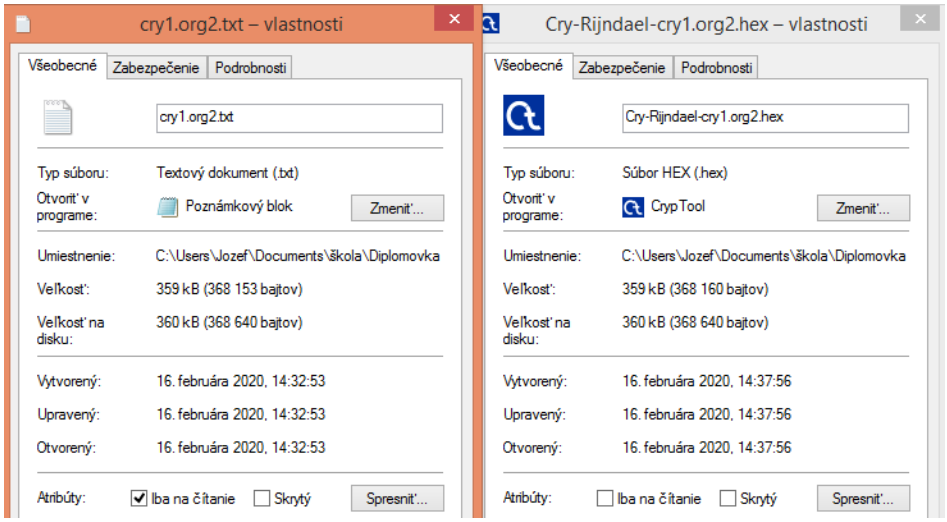


Obrázok 30 Vlastnosti súborov druhej úlohy (vlastné)

Z výsledkov sme zistili, že pri šifrovaní textu so 150 000 znakmi sa rýchlosť šifrovania oproti prvej prípadovej úlohy nezmenila a aj tu je časová odozva minimálna. A to aj napriek tomu, že počet znakov šifrovaného textu sa v porovnaní s prvou úlohou zvýšil až o 245 440 znakov. Veľkosť na disku sa medzi pôvodným a šifrovaným textom podobne ako v prvej úlohe nezmenila. Z tohto vychádza, že aj druhá prípadová úloha prostredníctvom AES týmto softvérom je veľmi efektívna.

Výsledok tretej prípadovej úlohy prostredníctvom algoritmu AES softvérom CrypTool:

Poslednou prípadovou úlohou je zašifrovať text, ktorý obsahuje 300 000 znakov. Po spustení šifrovania sme zistili, že čas odozvy sa minimálne zvýšila. Oproti prvej a druhej prípadovej úlohe, kde sa okamžite objavil zašifrovaný text, v tretej úlohe sa objavil s minimálnym časovým odstupom. Šifrovaný text, ktorý vznikol dosahuje až 736 320 znakov, čo je o 436 320 znakov viac ako pôvodný. Dešifrovanie tohto zašifrovaného textu nadobúda rovnaký čas odozvy ako jeho zašifrovanie. Čo sa týka jeho veľkosti, tak je to podobné ako v predchádzajúcich prípadových úlohách. Keďže je počet znakov 2 krát väčší ako v druhej prípadovej úlohe, jeho veľkosť na disku sa tiež zväčší dvakrát, čo znamená, že táto veľkosť bude 360 kB pre pôvodný aj šifrovaný text. Konkrétna veľkosť textového súboru pôvodného textu je 368 153 bajtov a veľkosť súboru šifrovaného textu je 368 160 bajtov. Znovu sa ukázalo, že aj napriek tomu, že je tam ovela viac znakov ako pri pôvodnom texte, veľkosťou sa však líšia len o 7 bajtov.



Obrázok 31 Vlastnosti súborov tretej úlohy (vlastné)

Výsledky nám jednoznačne ukázali, že pri našej tretej prípadovej úlohe s 300 000 znakmi sa jemne zvýšil čas odozvy oproti prvým dvom úlohám. Napriek tomu, že toto zvýšenie sa dá počítať v stotinách, sme zistili, že taký vysoký počet znakov začína zaberať CrypToolu viac času. Veľkosť obidvoch súborov na disku zostala rovnaká, čiže 360kB. Pri zohľadnení všetkých skutočností musíme povedať, že je toto šifrovanie taktiež veľmi efektívne. Pri šifrovaní 300 000 znakov je niekoľko stotinový čas odozvy výborný.

Výsledok bonusovej prípadovej úlohy prostredníctvom algoritmu AES softvérom CrypTool:

Hoci sme už vykonali všetky tri stanovené prípadové úlohy, je zaujímavé zistiť, aké veľké textové súbory dokáže Cryptool zašifrovať.

Preto si na začiatok vyberieme text s 1 000 000 znakmi. Pri vkladaní textu do poľa pre šifrovanie trvalo CrypToolu 4,68 sekúnd, kým to vzalo. Šifrovanie a dešifrovanie samo o sebe trvalo tak, ako pri tretej prípadovej úlohe.

Ďalšou úlohou budeme šifrovať text s 2 000 000 znakmi. S týmto počtom už začína mať CrypTool problémy. Celkové vloženie tohto textu do poľa trvalo 14,64 sekúnd medzi ktorými program nereagoval. Zašifrovanie tohto textu potom opäť trvalo rovnako ako pri tretej úlohe.

Poslednou úlohou skúsime zašifrovať text so 6 000 000 znakmi. Pri tejto veľkosti už začínajú mať problémy aj iné programy, ako napríklad word, ktorý sa strašne spomalil a pri každej akcii prechádza do stavu „nereaguje“. Po 36,15 sekunde sa načítal text do pola a vykonalo sa zašifrovanie.

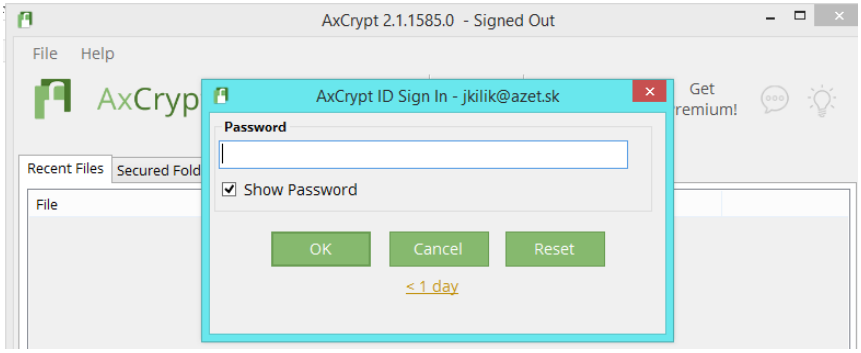
Z týchto pokusov sme zistili, že softvér CryptTool je výborný šifrovací softvér, ktorý dokáže rýchlo a efektívne šifrovať aj veľmi rozsiahle texty, ktoré iné programy už nemusia zvládať.

AxCrypt

Ďalším softvérom, ktorým budeme šifrovať prostredníctvom algoritmu AES tri prípadové úlohy na nazýva **AxCrypt**. Ako sme si už povedali, tento šifrovací softvér je vybavený šifrovacím algoritmom AES so 128 alebo 256 bitovým kľúčom a dokáže šifrovať

súbor, celú zložku alebo skupinu súborov. Umožňuje prideľovať heslá k zašifrovaným súborom a ich otváranie pre iných používateľov pomocou ich vlastného hesla.

Pri spustení tohto softvéru sa treba najprv prihlásiť a zadať heslo, ktoré bude vstupné heslo pre zašifrované súbory.

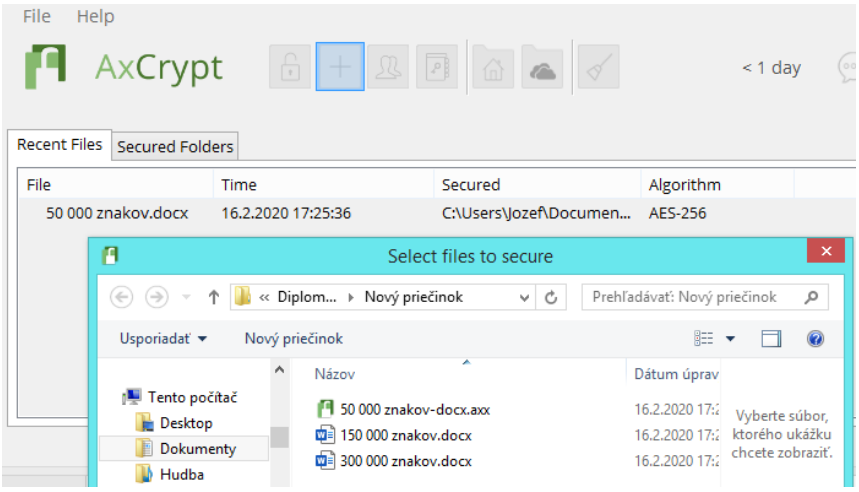


Obrázok 32 Vloženie hesla (vlastné)

Po zadaní hesla môžeme prejsť k šifrovaniu prípadových úloh.

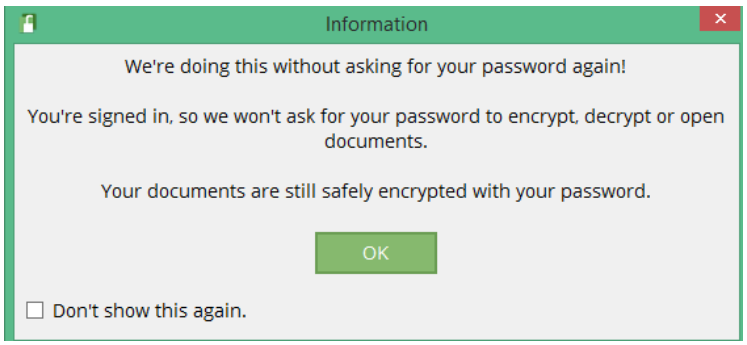
Výsledok prvej prípadovej úlohy prostredníctvom algoritmu AES softvérom AxCrypt:

V prvej prípadovej úlohu použijeme súbor s koncovkou docx s 50 000 znakmi. V softvéri si pomocou možnosti výberu vyberieme word prvej úlohy a zašifrujeme. Čas odozvy je okamžitý po kliknutí a vytvorí sa zašifrovaný súbor s koncovkou axx.



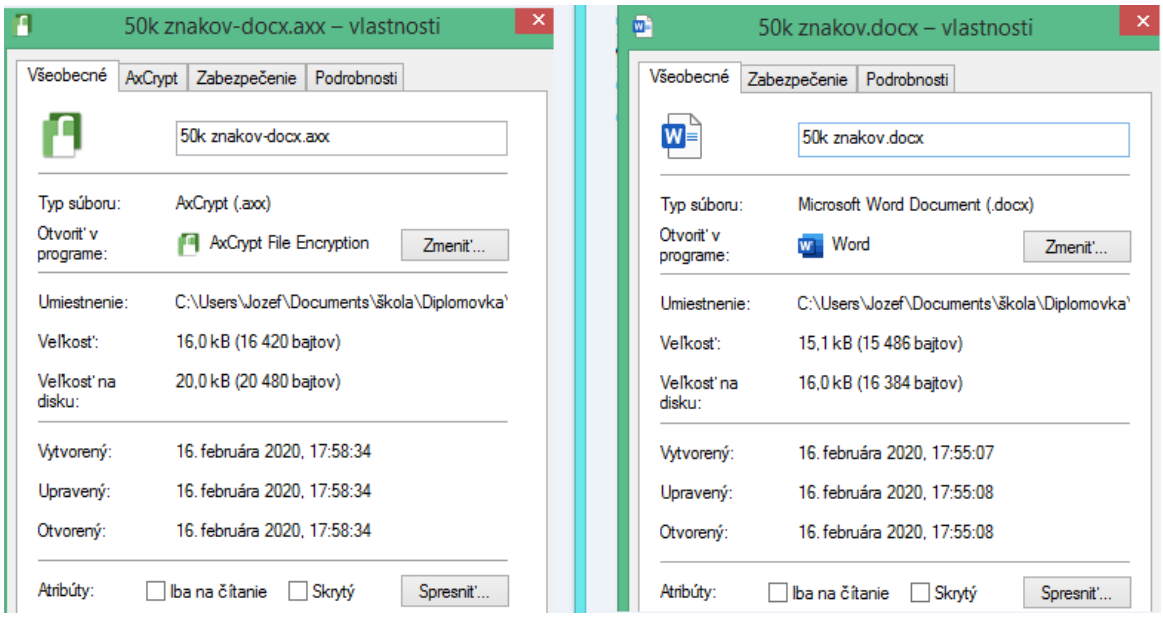
Obrázok 33 Šifrovanie v AxCrypt (vlastné)

Softvér si zapamätá všetky vykonané šifrovania a ponúka ich dešifrovanie. Pri snahe o otvorenie súboru si AxCrypt pýta heslo. V prípade, keď sme sa už prihlásili do AxCryptu predtým nám vypíše informáciu o tom, že už nemusíme znovu písať heslo.



Obrázok 34 Informácia o hesle (vlastné)

Čo sa týka porovnania veľkosti súborov, tak ako si môžeme v nasledujúcom obrázku všimnúť, veľkosť pôvodného súboru je 15 486 bajtov a veľkosť zašifrovaného súboru stúpol na 16 420 bajtov, čo znamená, že rozdiel medzi týmito súbormi je 934 bajtov. Keď sa však pozrieme koľko miesta zaberajú na disku, tak veľkosť pôvodného súboru zaberá 16kB a šifrovaného súboru o 4 kB viac, čiže 20kB.

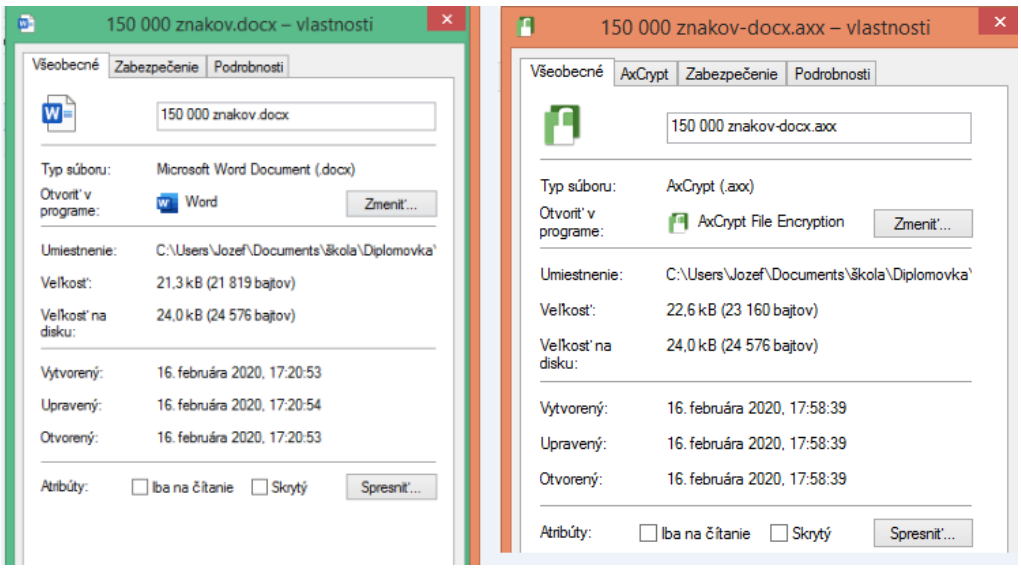


Obrázok 35 Porovnanie súborov prvej úlohy (vlastné)

Pri dešifrovaní šifrovaného textu si AxCrypt znovu pýta heslo, bez ktorého dešifrovanie nie je možné. Po zadaní správneho hesla je časová odozva okamžitá. Zo získaných výsledkov sme zistili, že časová odozva je okamžitá, rozdiel medzi veľkosťou na disku pôvodného a šifrovaného súboru je 4 kB a na vykonávanie zmien s týmto šifrovaným súborom je potrebné heslo. Z tohto vychádza, že prvá prípadová úloha je vysoko efektívna na šifrovanie.

Výsledok druhej prípadovej úlohy prostredníctvom algoritmu AES softvérom AxCrypt:

Pre realizáciu druhej prípadovej úlohy použijeme súbor s koncovkou docx s 100 000 znakmi. Po výbere zvoleného wordovského súboru s týmto počtom znakov ho prostredníctvom softvéru zašifrujeme. Časová odozva tejto úlohy je 1 sekunda, čo znamená, že oproti prvej prípadovej úlohy, kde bola okamžitá, sa spomalila. Taktiež ako v prvej úlohe, aj tu je dôležité poznať heslo na základe ktorého je súbor zašifrovaný, inak nie je možné tento súbor otvoriť alebo dešifrovať. Veľkosť obyčajného súboru so 150 000 znakmi je 21 819 bajtov a veľkosť šifrovaného súboru sa oproti nešifrovanému zvýšila o 1 341 bajtov na 23 160 bajtov. Z pohľadu veľkosti na disku sa nám veľkosť vôbec nezmenila, pretože aj pôvodný aj šifrovaný súbor zaberajú 24 kB.

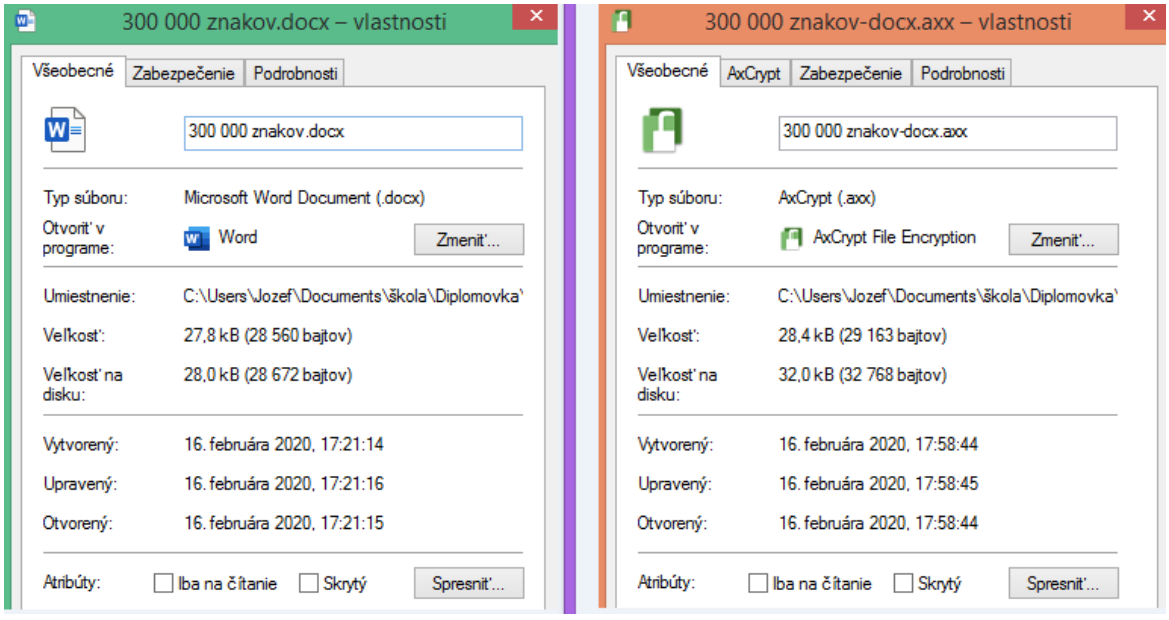


Obrázok 36 Porovnanie súborov druhej úlohy (vlastné)

Na základe výsledkov druhej prípadovej úlohy sme zistili, že časová odozva od kliknutia sa zvýšila na 1 sekundu, veľkosť súborov na disku je zhodný a heslo je podobne ako pri prvej úlohe veľmi dôležité. Aj napriek tomu, že sa časová odozva zvýšila je šifrovanie tejto úlohy efektívne.

Výsledok tretej prípadovej úlohy prostredníctvom algoritmu AES softvérom AxCrypt:

V tretej prípadovej úlohe budeme šifrovať docx súbor s 300 000 znakmi. Po nastavení hesla si takisto ako v predchádzajúcich úlohách vyberieme súbor, ktorý budeme šifrovať. Časová odozva od začatia šifrovania je 1 sekunda, čo je úplne rovnaká ako v druhej prípadovej úlohe. To znamená, že softvér šifruje súbor s počtom znakov od 150 000 po 300 000 rovnakou rýchlosťou. Vieme povedať, že veľkosť pôvodného súboru dosahuje 28 560 bajtov a šifrovaného súboru 29 163 bajtov, čo je o 603 viac. Keď si porovnáme tento rozdiel s ostatnými prípadovými úlohami, tak sme zistili, že pri tejto úlohe je rozdiel najmenší. Ďalším zaujímavým zistením je, že pôvodný súbor zaberá na disku veľkosť 28kB, no šifrovaný zaberá 32kB. V porovnaní z predchádzajúcimi úlohami je to rovnaký rozdiel ako v prvej úlohe, no v druhej úlohe boli tieto veľkosti rovnaké.



Obrázok 37 Porovnanie súborov tretej úlohy (vlastné)

Pri získaných výsledkov sme zistili, že v tretej prípadovej úlohe je veľkosť šifrovaného súboru väčšia o 4kB ako pri pôvodnom súbore, časová odozva je 1 sekunda

a podobne ako pri všetkých úlohách tohto softvéru je dôležité heslo. Z týchto poznatkov môžeme povedať, že z časového hľadiska je veľmi efektívna oproti druhej prípadovej úlohe, keďže časová odozva je rovnaká, no počet znakov je dvakrát väčší. Celkovo môžeme povedať o tretej prípadovej úlohe ako vysoko efektívnej pre šifrovanie.

Výsledok bonusovej prípadovej úlohy prostredníctvom algoritmu AES softvérom AxCrypt:

Podobne ako pri softvéri Cryptool, kde sme hľadali, aké veľké textové súbory dokáže zašifrovať, sa budeme zaoberať, aké velké word súbory dokáže softvér AxCrypt zašifrovať. Znovu si vyskúšame súbory s počtom znakov 1 000 000, 2 000 000 a 6 000 000 a dozvieme sa aké efektívne šifrovanie ponúka tento softvér.

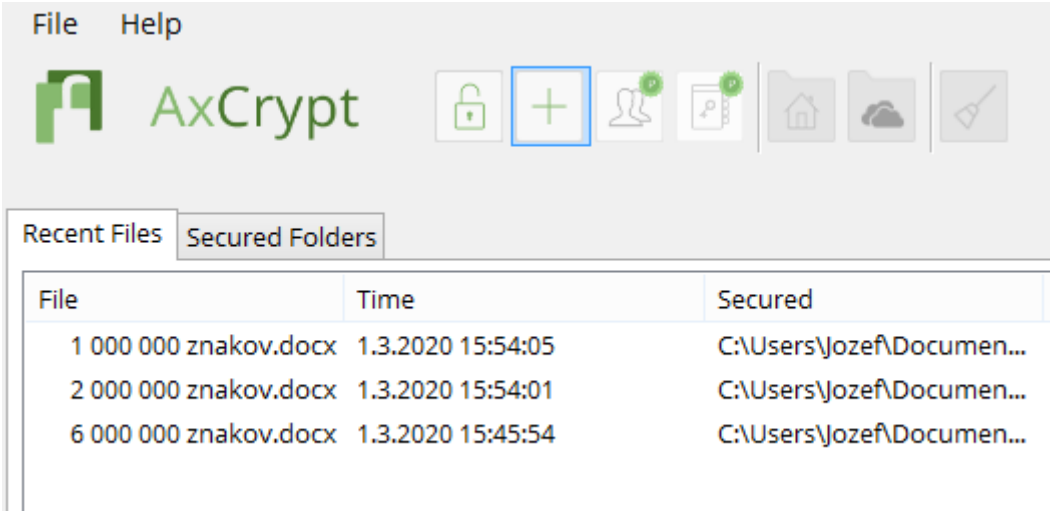
Začneme najprv s 1 000 000 počtom znakom. Časová odozva zašifrovania tejto úlohy je 2 sekundy. Pôvodný súbor na disku zaberá 48 kB, čo je v porovnaní s veľkosťou šifrovaného súboru väčšie o 12kB. V tejto úlohe dosahuje šifrovaný súbor veľkosť na disku len 32 kB. Tu vidíme zaujímavú zmenu, keďže pri prípadových úlohách boli veľkosti na disku prevažne rovnaké. Pri zhodnotení tejto úlohy sme zistili, že softvér efektívne zvláda šifrovať súbory aj takejto veľkosti.

Ďalšou bonusovou úlohou je súbor s počtom znakov 2 000 000. Časová odozva je úplne rovnaká ako pri predchádzajúcej úlohe s 1 000 000 znakmi. Veľkosť na disku pôvodného súboru dosahuje 64 kB a šifrovacieho súboru je 24 kB. Podobne ako pri 1 000 000 znakoch aj tu je veľkosť šifrovaného súboru nižšia a tento rozdiel dosahuje až 40 kB. Na základe týchto zistení vychádza, že čím je vyšší počet znakov a tým pádom veľkosť docx súboru, tým menší bude šifrovaný súbor.

Poslednú úlohu, ktorú budeme testovať a overovať zistenia je súbor so 6 000 000 znakmi. Po kliknutí šifrovania nám softvér zašifruje súbor za 2 sekundy, čo je identické číslo ako v predchádzajúcich dvoch prípadoch. Keď sa pozrieme na veľkosti na disku, vychádza nám, že pôvodný súbor zaberá miesto 164 kB a šifrovaný 32 kB, čo je o 132 kB menej.

Prostredníctvom týchto bonusových úloh sme zistili, že softvér AxCrypt zmenšuje veľkosť pôvodného súboru šifrovaním. Čím má tento pôvodný súbor vyššiu veľkosť, tým je veľkosť šifrovaného nižšia a rozdiel medzi nimi sa zvyšuje. Preto môžeme zhodnotiť softvér AxCrypt ako veľmi efektívny hlavne pre súbory s vyššou veľkosťou a väčším počtom

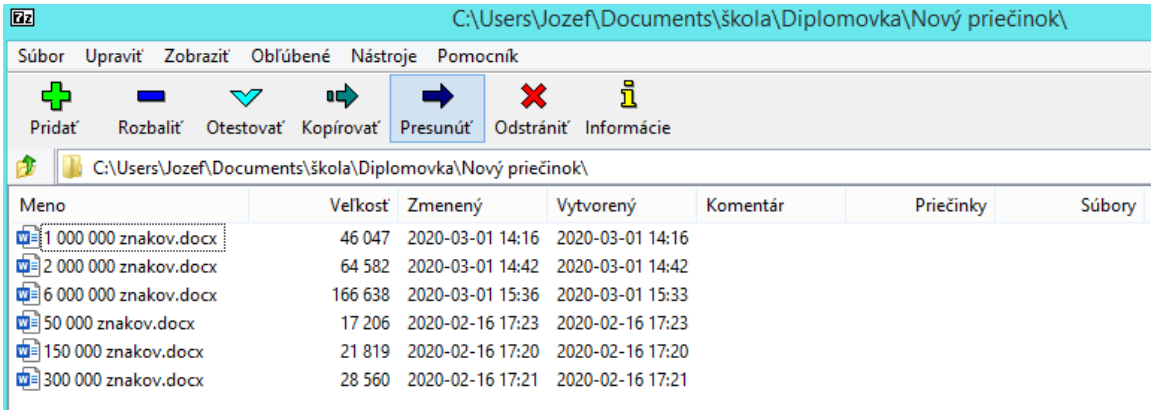
znakov. Časová odozva softvéru je na výbornej úrovni, pretože je minimálna a šifrovaný súbor vznikne v priebehu pár sekúnd.



Obrázok 38 Prehľad šifrovaných súborov (vlastné)

7-Zip File Manager

Tretím softvérom, ktorý budeme testovať a hodnotiť jeho efektivitu šifrovania je softvér 7-Zip File Manager. Umožňuje extrahovať väčšinu archívov a používa šifru AES-256. Podobne ako AxCrypt nešifruje celý pevný disk, ale iba konkrétne súbory a dokumenty, v ktorých sa nachádzajú citlivé informácie. Na nasledujúcom obrázku vidíme vzhľad tohto softvéru, kde je otvorený priečinok so súbormi jednotlivých prípadových úloh, ktoré budeme testovať.



Obrázok 39 Vzhľad 7-Zip File Manager (vlastné)

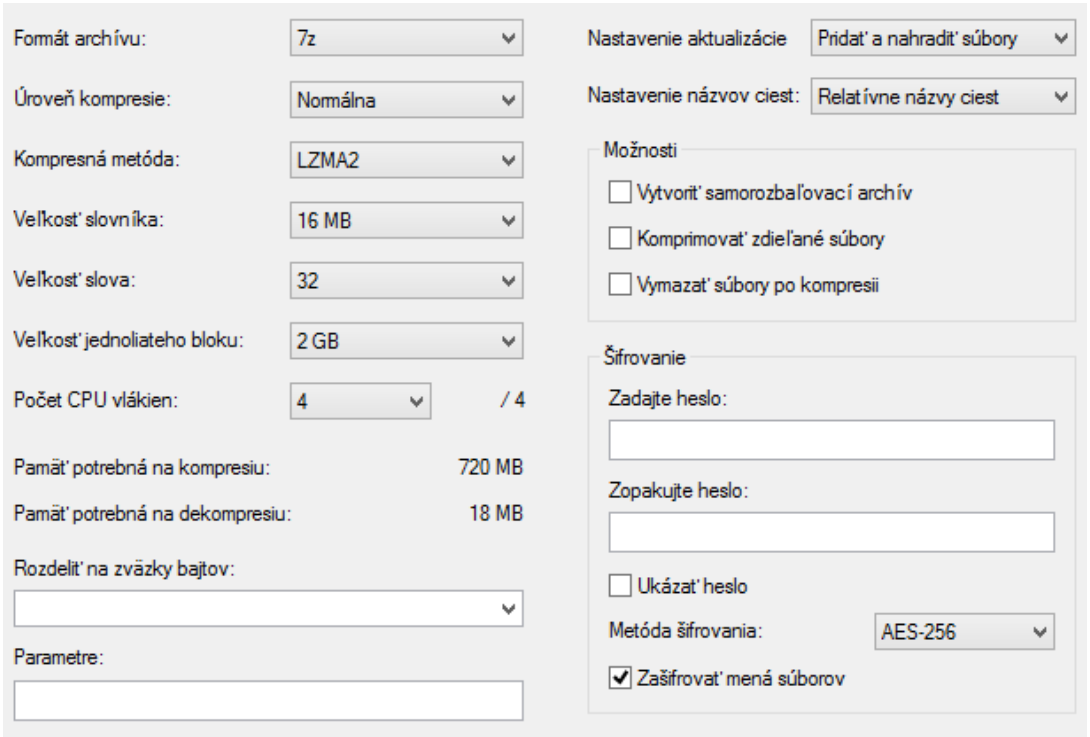
Na jednotlivé zašifrovanie musíme vybrať vhodné nastavenia, ktoré použijeme. Na základe týchto nastavení bude softvér šifrovať.

Medzi tieto nastavenia patrí formát archívu, ktorý ponúka rôzne možnosti archivácie nášho súboru. Predvolený formát a formát, ktorý budeme používať na šifrovanie je 7z, no je možné si vybrať aj bzip2, gzip, tar, wim, xz alebo zip.

Ďalším potrebným nastavením je úroveň kompresie, ktorá umožňuje vybrať si možnosti bez kompresie, najrýchlejšia, rýchla, normálna, maximálna a ultra. Pri našich prípadových úlohách použijeme normálnu kompresiu.

Nasledujúcimi nastaveniami sú nastavenia veľkostí slovníka, slova a jednoliateho bloku. Veľkosť slovníka je ohraničená hodnotami 64 kB až 64 MB. Veľkosť slova je možné určiť medzi 8 a 273. Poslednú veľkosť je umožnené nastaviť na nejednoliatu, jednoliatu alebo v intervale 1 MB až 64 GB. Na testovanie našich úloh si nastavíme veľkosť slovníka na 16 MB, veľkosť slova na 32 a veľkosť jednoliateho bloku na 2 GB.

Podobne ako pri softvéri AxCrypt, aj tu je dôležité zadať bezpečné heslo, pod ktorým sa vykoná šifrovanie súboru. Týmto heslom sa potom dajú tieto šifrované súbory otvoriť alebo odšifrovať.

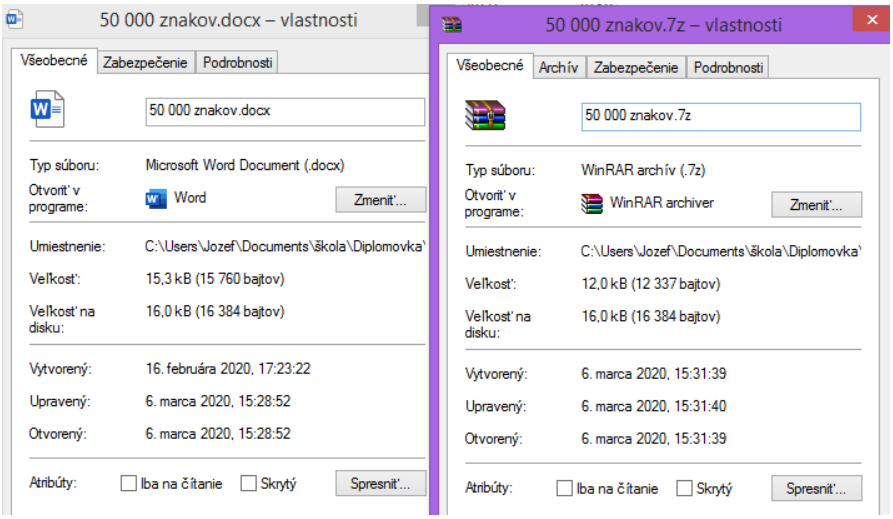


Obrázok 40 Nastavenie sifrovania 7-Zip

Po nastavení všetkých dôležitých nastavení je možné šifrovať súbory a zisťovať jednotlivé náležitosti tohto procesu.

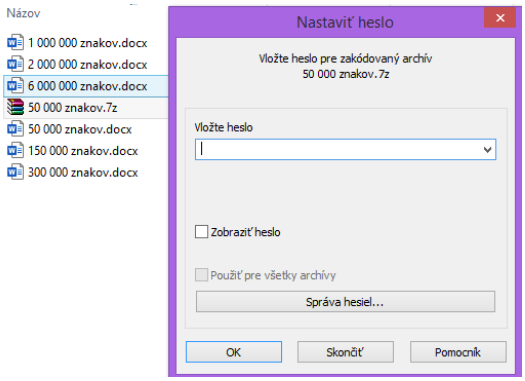
Výsledok prvej prípadovej úlohy prostredníctvom algoritmu AES softvérom 7-ZIP File Manager:

V prvej prípadovej úlohe si vyberieme word dokument s veľkosťou 50 000 znakov a nastavíme detaily šifrovania. Pri nastavovaní nastavíme heslo, ktoré je dôležité pre ďalšiu prácu so zašifrovaným súborom. Po spustení šifrovania sme zistili, že časová odozva tohto softvéru v tejto prípadovej úlohe je 1 sekunda. Veľkosť pôvodného súboru je podobne ako pri predchádzajúcej prvej úlohe iného softvéru 15 486 bajtov. Veľkosť zašifrovaného súboru je 12 337 bajtov, čo znamená, že po zašifrovaní sa zmenšila o 3 149 bajtov. Keď sa však pozrieme na veľkosť týchto dvoch súborov na disku, tak zistíme, že je rovnaká. Obidva súbory zaberú zhodne 16 kB. Z pôvodného súboru vznikne zašifrovaný balík s koncovkou 7z. Tento balík je teraz možné rozbaliť cez WinRaR programe len za pomoci nastaveného hesla.



Obrázok 41 Porovnanie súborov prvej úlohy (vlastné)

Zadaním správneho hesla sa tento súbor odšifruje rovnako ako obyčajný balíkový súbor.

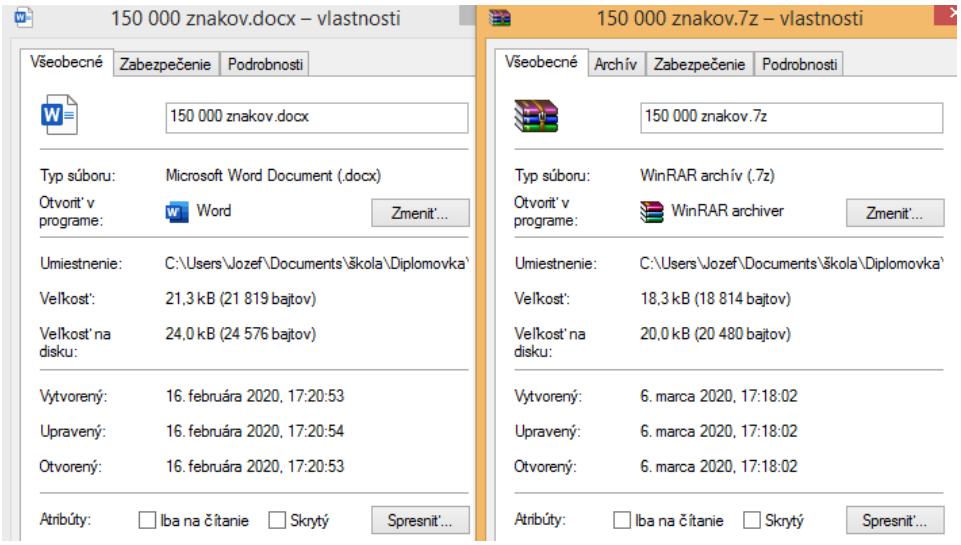


Obrázok 42 Zadanie bezpečnostného hesla

Pri šifrovaní tejto prvej prípadovej úlohy sme dostali výsledky na základe, ktorých vieme povedať, že šifrovanie súboru týmto softvérom zmenší veľkosť pôvodného súboru, no veľkosť na disku zostane rovnaká. Takisto sme zistili, že časová odozva je rýchla, čo celkovo znamená, že šifrovanie prvej úlohy zvláda 7-ZIP FILE efektívne.

Výsledok druhej prípadovej úlohy prostredníctvom algoritmu AES softvérom 7-ZIP File Manager:

V druhej prípadovej úlohe budeme používať súbor so 150 000 znakmi a po nastavení všetkých náležitostí vrátane hesla ho zašifrujeme. Čas, za ktorý zašifrovanie prebehne je 1 sekunda, podobne ako pri súbore z prvej úlohy pri 50 000 znakoch. Veľkosť pôvodného súboru je 21 819 bajtov a v porovnaní so zašifrovaným súborom, ktorého veľkosť je 18 814 bajtov, sa potvrdzuje zistenie z prvej prípadovej úlohy, kde platí, že šifrovanie prostredníctvom softvéru 7-ZIP zmenšuje veľkosť pôvodného súboru. V tomto prípade je veľkosť zmenšená o 3 005 bajtov. Na druhej strane sa však v tejto prípadovej úlohe nerovnajú veľkosti súbory na disku, ako tomu bolo v predchádzajúcej, keďže táto veľkosť pôvodného súboru zaberá 24 kB a šifrovaného len 20 kB.



Obrázok 43 Porovnanie súborov druhej úlohy (vlastné)

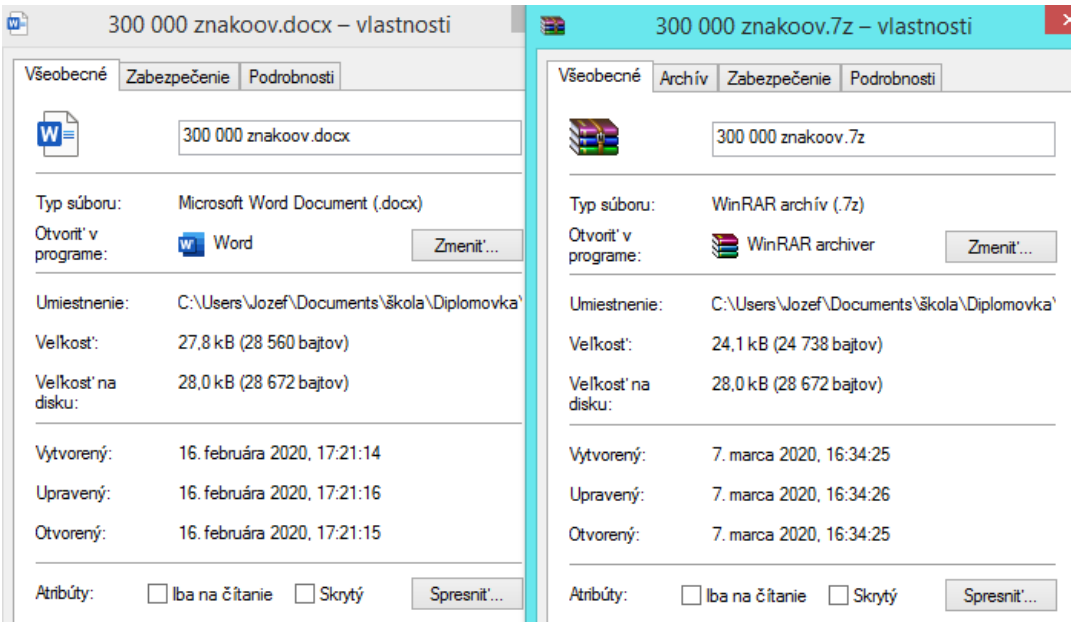
Rovnako ako v prvej úlohe budeme na odšifrovanie tohto šifrovaného súboru potrebovať heslo, ktoré sme zadali pri nastaveniach šifrovania. Bez neho sa ku tomuto súboru nikto nedostane a nezistí, čo sa v tomto balíku nachádza.

Zo zistení môžeme povedať, že šifrovanie druhej prípadovej úlohy je efektívne, pretože jeho časová odozva je rovnaká ako pri prvej prípadovej úlohe, veľkosť šifrovaného balíka sa zmenší a dokonca sa zmenší aj jeho veľkosť na disku.

Výsledok tretej prípadovej úlohy prostredníctvom algoritmu AES softvérom 7-ZIP File Manager:

Pri tretej prípadovej úlohe vyberieme v softvéri na šifrovanie súbor s 300 000 znakmi, ktorému opätovne nastavíme dohodnuté nastavenia a heslo, s ktorým budeme vedieť tento šifrovaný súbor odšifrovať. Po spustení šifrovania sme zistili, že aj v tretej prípadovej úlohe je časová odozva šifrovania rovnaká ako v predchádzajúcich dvoch prípadoch 1 sekunda. Z tohto nám vychádza, že softvér zvláda šifrovať bez problémov súbory s počtom znakov do 300 000. Veľkosť pôvodného súboru má 28 560 bajtov a veľkosť šifrovaného súboru sa opäť znížila a dosahuje 24 738 bajtov. Týmto sa nám úplne potvrdilo tvrdenie, ktoré sme objavili pri druhej prípadovej úlohe, že softvér 7-Zip prostredníctvom zašifrovania súboru a následné vytvorenie šifrovaného balíka zmenšuje veľkosť pôvodného súboru. Odšifrovanie súboru spôsobí, že sa jeho veľkosť vráti na svoju

pôvodnú, akú mala pred šifrovaním. Keď sa pozrieme na veľkosť na disku zistíme, že pôvodný súbor zaberá na disku 28 kB, čo je rovnaké ako aj šifrovaný. Tu nastala zmena v porovnaní s druhou úlohou, kde sa tieto dve veľkosti nerovnali a súhlasí to s výsledkami prvej prípadovej úlohy.



Obrázok 44 Porovnanie súborov tretej úlohy (vlastné)

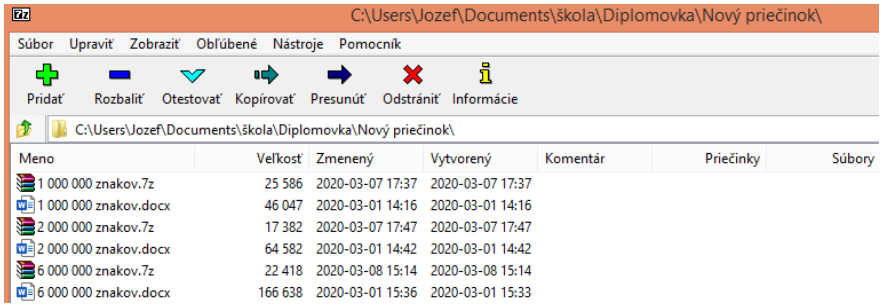
Zo získaných výsledkov sme sa dozvedeli o tom, že vo všetkých troch prípadoch dosahuje šifrovaný balík menšiu veľkosť ako pôvodný, dozvedeli sme sa taktiež, že pri všetkých troch úlohách bola časová odozva rovnaká. Ďalšou informáciou, ktorú sme získali je, že veľkosť na disku pôvodného súboru je v prvej a tretej úlohe rovnaká ako šifrovaného, no v druhej úlohe vyššia. Na základe zistení môžeme povedať, že všetky tri úlohy sú efektívne pre šifrovanie.

Výsledok bonusovej prípadovej úlohy prostredníctvom algoritmu ARS softvérom 7-ZIP File Manager:

Na základe výsledkov troch prípadových úloh sme zistili, že sú efektívne, preto znovu ako pri iných softvéroch, aj tu budeme zisťovať do akej výšky znakov bude 7-Zip File manager šifrovať efektívne. Prvou bonusovou úlohou bude zašifrovať súbor s počtom znakov 1 000 000. Pri tomto počte znakov sa časová odozva nezvýšila a rovnako ako v predchádzajúcich úlohách je 1 sekunda. Veľkosť súborov spĺňa získané poznatky o tom, že veľkosť šifrovaného súboru je menšia ako pôvodného. Čo však stojí za pozornosť je veľkosť na disku, pretože pôvodný súbor zaberá na disku o 20 kB viac ako šifrovaný súbor.

Druhou bonusovou úlohou budeme šifrovať súbor s 2 000 000. Časová odozva šifrovania s týmto počtom znakov je znovu 1 sekunda, čo je rovnaké ako všetky predchádzajúce prípady. Keď sa pozrieme na veľkosť súborov, znovu sa potvrdzuje pravidlo, že pôvodný súbor je väčší ako šifrovaný. Z pozorovania a skúmania zisťujeme , že čím je pôvodný súbor vačší, tým je väčší rozdiel veľkosti medzi ním a šifrovaným súborom. Podobne to platí aj pri veľkosti na disku. Pri tomto počte znakov šifrovaný súbor zaberá miesto na disku o 44 kB menej ako pôvodný súbor.

Posledná bonusová úloha, ktorú budeme testovať je šifrovanie súboru s veľkosťou 6 000 000 znakov. Súbor s takouto veľkosťou počtu znakov sa zašifruje za 2 sekundy. To znamená, že jeho časová odozva oproti predchádzajúcim prípadom sa zvýšila o 1 sekundu. Veľkosť tohto pôvodného súboru má takmer 8 krát viac bajtov ako vzniknutý zašifrovaný súbor. Z tohto sme si definitívne potvrdili výrok, že okrem zmenšenia veľkosti šifrovaného súboru oproti pôvodnému, záleží aj na tom, aký veľký je pôvodný súbor. To znamená, že čím väčšiu veľkosť má, tým väčší rozdiel bude medzi veľkosťami pôvodného a šifrovaného súboru. Keď sa pozriem na veľkosť súborov, ktoré zaberajú na disku, tak zistíme, že pôvodný súbor zaberá na disku o 140 kB viac ako šifrovaný.



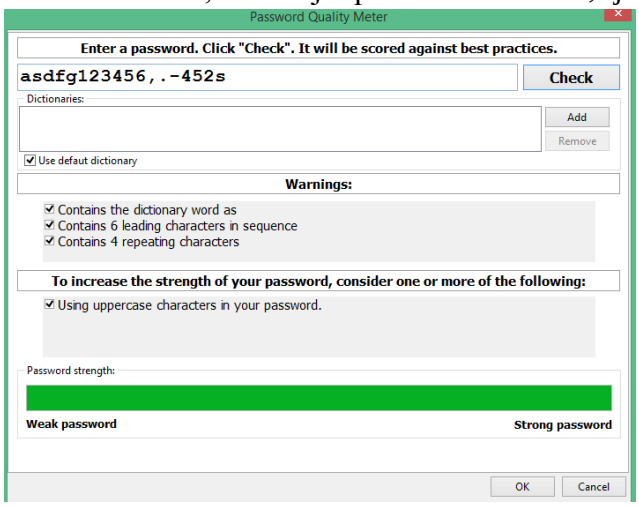
Obrázok 45 Prehľad bonusových úloh (vlastné)

Secure IT

Ďalším softvérom, ktorý budeme používať a testovať jeho efektívnosť, je Security IT slúžiaci na šifrovanie súborov a umožňuje komprimovanie súborov. Na prístup ku ktorémukoľvek zašifrovanému súboru vyžaduje dostatočné silné heslo. Secure IT umožňuje šifrovať pomocou dvoch algoritmov AES alebo BLOWFISH. My budeme v tejto časti práce pracovať s algoritmom AES s 256 bitovým kľúčom. Na zašifrovanie jednotlivých súborov si najprv musíme vybrať konkrétny dokument. Po splnení tejto požiadavky môžeme kliknúť na tlačítko „encrypt“, ktoré nám otvorí nové okno s nastaveniami šifrovania. Tu si zadáme

heslo, ktorým zabezpečíme tento šifrovaný súbor a bez ktorého ho nebude možné odšifrovať. Softvér na základe vloženia hesla zhodnotí, či je toto heslo dostatočne silné, pričom vieme jeho parametre prostredníctvom tlačidla „Password Quality Meter“ bližšie špecifikovať. Okrem hesla nastavíme aj kompresiu, ktorej hodnotu môžeme nastaviť na minimalizáciu času, minimalizíciu veľkosti, normálnu alebo žiadnu kompresiu. My si pre naše šifrovania vyberieme kompresiu s minimalizáciou času. Posledné nastavenie, ktoré je potrebné nastaviť, je algoritmus, ktorým budeme súbory šifrovať. Ako sme už spomínali vyššie, v našej práci si určíme algoritmus AES.

Obrázok 46 Kontrola hesla (vlastné)



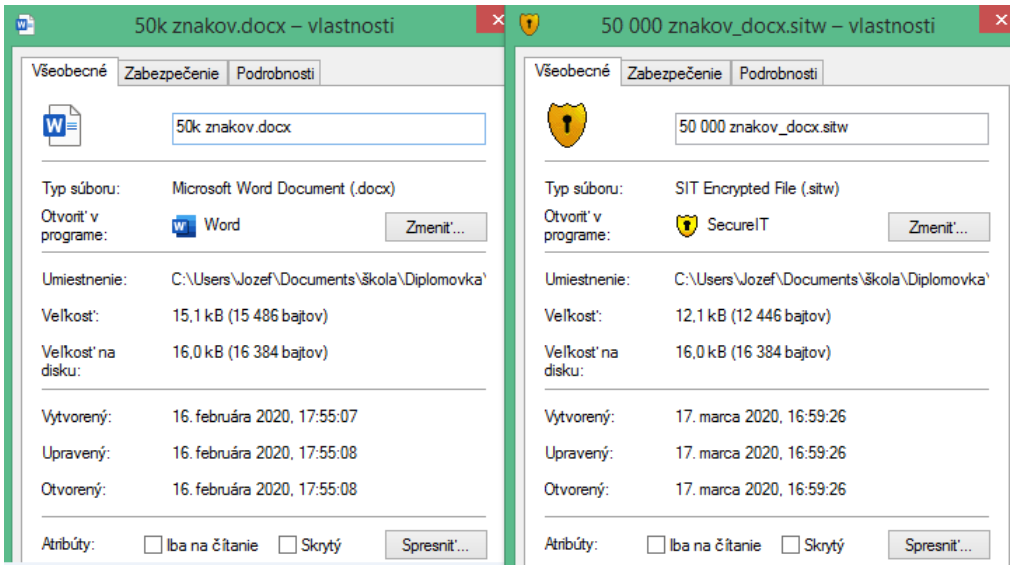
Obrázok 47 Nastavenie šifrovania pri algoritme SecureIT (vlastné)

Po nastavení všetkého potrebného, pokračujeme cez tlačítko „next“ do ďalšieho okna nastavení, kde si určíme, kam tento zašifrovaný súbor uložíme. Taktiež si môžeme ponastavovať ďalšie možnosti, ktorými nám softvér zjednoduší prácu. Potom ako, sme vyplnili všetko, ako sme chceli, môžeme znovu stlačiť na tlačítko „next“, po ktorom nám začne softvér šifrovať a vznikne nový zašifrovaný súbor.

Obrázok 48 Nastavenia softvéru SecureIT (vlastné)

Výsledok prvej prípadovej úlohy prostredníctvom algoritmu AES softvérom Secure IT:

V prvej prípadovej úlohe budeme šifrovať, podobne ako pri predchádzajúcich algoritmoch, súbor s príponou .docx s počtom znakov 50 000. Vyberieme si tento dokument v softvéri a po nastavení všetkých potrebných údajov ho môžeme zašifrovať. Zašifrovanie prebehlo okamžite po potvrdení nastavenia a spustenia šifrovania. To znamená, že čas, za ktorý sa toto šifrovanie vykonalo, je minimálny a veľmi efektívny. Keď sa pozrieme na porovnanie pôvodného súboru a novovzniknutého zašifrovaného súboru, vidíme, že pôvodný dosahuje veľkosť 15 486 bajtov a šifrovaný 12 446 bajtov. Z tohto sme zistili, že šifrovaný súbor má o 3 040 bajtov menšiu veľkosť ako pôvodný. Keď si však všimneme veľkosť týchto súborov, ktorú zaberajú na disku, zistíme, že u obidvoch súboroch je zhodná. Môžeme teda povedať, že aj napriek veľkostným rozdielom súborov je ich disková veľkosť rovnaká a dosahuje 16 kB.



Obrázok 49 Vlastnosti súborov (vlastné)

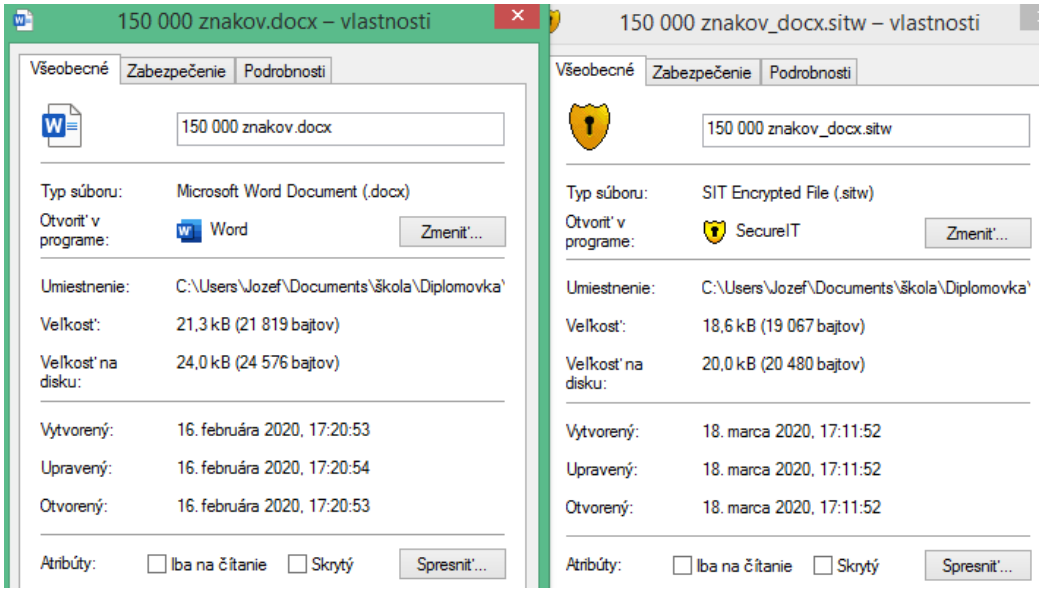
Keď chceme odšifrovať tento súbor, musíme znovu zadať heslo, ktoré sme nastavili pri jeho šifrovaní. V prípade straty tohto hesla je možné spomenúť si naň za pomoci rady, ktorú sme si mohli napísať pri začiatočnom nastavovaní. No ak aj napriek tomu je heslo stratené, novovzniknutý šifrovaný súbor nebude možné odšifrovať ani otvoriť. Pokiaľ však

heslo vieme, softvér nám tento šifrovaný súbor odšifruje. Následne vznikne odšifrovaný priečinok, v ktorom sa nachádza pôvodný dokument.

Na základe získaných výsledkov vieme, že čas šifrovania je okamžitý, veľkosť nového zašifrovaného súboru je menšia ako pôvodného a veľkosť na disku je rovnaká. Z tohto môžeme povedať, že prvá prípadová úloha je efektívna.

Výsledok druhej prípadovej úlohy prostredníctvom algoritmu AES softvérom Secure IT:

V tejto prípadovej úlohe budeme skúmať šifrovanie dokumentu s príponou .docx s veľkosťou 150 000 znakov. Po nastavení všetkých potrebných údajov môžeme zistiť, ako efektívne softvér zašifruje tento súbor. Pri spustení šifrovania sa nám výsledok zobrazí okamžite. To znamená, že čas, za ktorý sa šifrovanie vykoná, je rovnaký ako čas, ktorému to trvalo pri prvej prípadovej úlohe. Veľkosť pôvodného súboru je, podobne ako v predchádzajúcich softvéroch tejto prípadovej úlohy, 21 819 bajtov. Keď si to porovnáme s veľkosťou novovzniknutého zašifrovaného súboru, ktorý má 19 067 bajtov, tak je táto veľkosť menšia o 2 752 bajtov. Znovu sa potvrdilo, že softvér Secure IT zmenšuje šifrovaním pôvodný súbor. Pokiaľ si všimneme miesto, ktoré zaberajú na disku, tak je tam zmena. V prvej prípadovej úlohe sme zistili, že miesta na disku zaberajú zhodne, no v tejto prípadovej úlohe to už neplatí. Veľkosť pôvodného súboru na disku dosahuje 24 kB, no šifrovaného súboru len 20 kB.



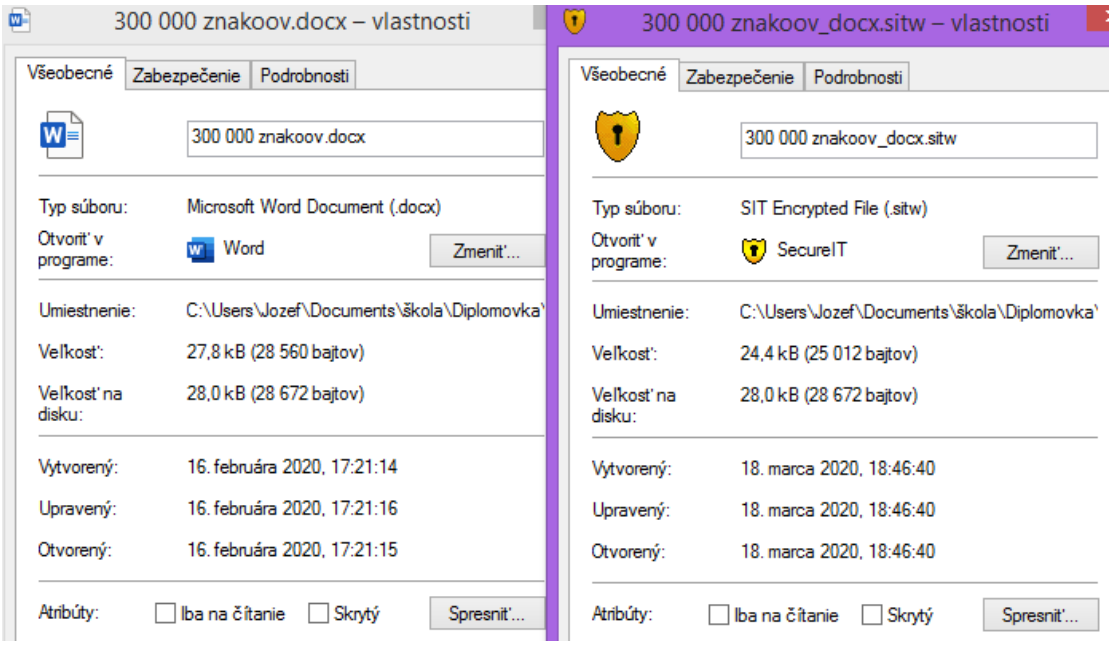
Obrázok 50 Vlastnosti súborov (vlastné)

Na odšifrovanie tohto zašifrovaného súboru potrebujeme znovu použiť heslo, ktoré sme si zadali na začiatku nastavovania. Odšifrovanie prebehlo okamžite a pôvodný súbor je s nezmenenou veľkosťou použiteľný na otváranie.

Z tohto môžeme povedať, že druhá prípadová úloha je taktiež efektívna, lebo čas trvania šifrovania aj odšifrovania je minimálny a zašifrovaný súbor zaberá na disku menej miesta ako pôvodný súbor.

Výsledok tretej prípadovej úlohy prostredníctvom algoritmu AES softvérom Secure IT:

V tretej prípadovej úlohe sa budeme zaoberať šifrovaním dokumentu s príponou .docx s počtom znakov 300 000. Opäť si nastavíme všetky nastavenia, ktoré od nás softvér Secure IT žiada, aby mohol vykonať bezproblémové šifrovanie. Reakcia softvéru pri dokumente s týmto počtom znakov je taká istá ako v predchádzajúcich dvoch úlohách. Šifrovanie prebehne okamžite a následne otvorí okno s našim zašifrovaným súborom. Pôvodná veľkosť tohto súboru je 28 560 bajtov. Pri veľkosti šifrovaného súboru znovu platí, že je menší ako pôvodný a dosahuje 25 012 bajtov. Zaujímavé je však miesto na disku, ktoré tieto dva súbory zaberajú. Podobne ako v prvej prípadovej úlohe sa aj v tretej tieto veľkosti na disku rovnajú a dosahujú 28 kB, zatial čo v druhej prípadovej úlohe tieto veľkosti neboli zhodné.



Obrázok 51 Vlasnosti súborov (vlastné)

Odšifrovanie prebieha rovnako ako v predchádzajúcich úlohách prostredníctvom potrebného hesla. Čas, za ktorý sa zašifrovaný súbor odšifruje, je 1 sekunda. Tu vidíme, že čas odšifrovania sa zvýšil v porovnaní s prvou a druhou prípadovou úlohou.

Z výsledkov môžeme povedať, že softvér pri tomto počte znakov šifruje stále okamžite, no jeho odšifrovanie sa spomalilo na 1 sekundu. Taktiež platí, že veľkosť šifrovaného súboru je menší ako pôvodného súboru, no ich veľkosti na disku sa rovnajú. Efektivitu šifrovania môžeme určiť ako vysokú, pretože aj napriek spomalenému odšifrovaniu je šifrovanie stále expresne rýchle.

Výsledok bonusovej prípadovej úlohy prostredníctvom algoritmu AES softvérom Secure IT:

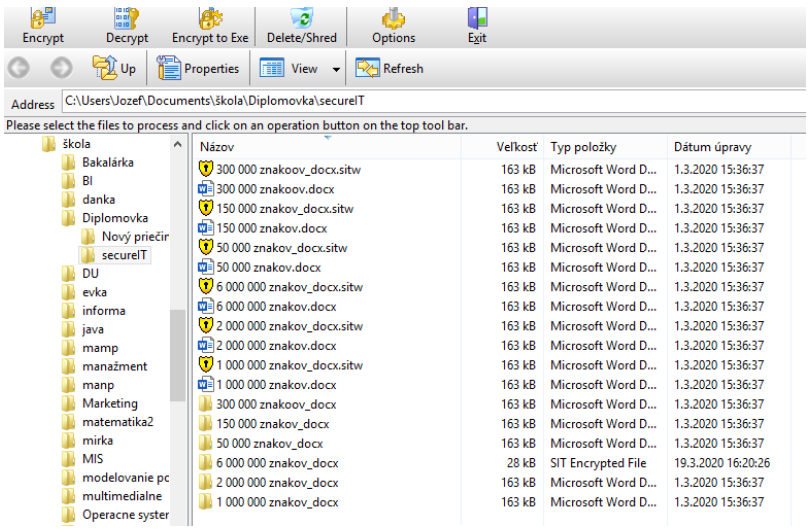
Po zistení prvých troch prípadoch sme zistili, že všetky úlohy zvláda softvér Seruce IT bez problémov a veľmi efetívne. Preto sa v bonusových úlohách pozrieme, ako bude reagovať tento softvér pri šifrovaní .docx dokumentov s počtom znakov 1 000 000, 2 000 000 a 6 000 000.

Prvou bonusovou úlohou bude šifrovanie súboru s 1 000 000 znakmi. Šifrovanie tejto úlohy s týmto počtom znakom prebehlo okamžite po kliknutí potvrdenia nastavení šifrovania a jeho spustenia. Zistili sme, že tento počet znakov v dokumente neuberá efektivitu šifrovania softvéru. Veľkosť pôvodného súboru je väčšia ako veľkosť nového zašifrovaného súboru, čo platí v každej prípadovej úlohe. Keď sa pozrieme na miesto týchto súborov, ktoré zaberajú na disku, tak zistíme, že šifrovaný súbor zaberá o 20 kB menej ako pôvodný. Podľa pozorovaní zisťujeme, že v niektorých prípadoch boli tieto veľkosti zhodné a niekde bola veľkosť šifrovaného dokumentu nižšia. Rýchlosť odšifrovania sa zvýšila na 2 sekundy, od potvrdenia tlačítka, ktoré tento proces vykonáva. Celkovo je šifrovanie a odšifrovanie tejto úlohy, vzhľadom na počet znakov a veľkosť, ktorú zaberá šifrovaný súbor na disku, efektívne.

V druhej bonusovej úlohe sa budeme venovať šifrovaniu súboru s príponou .docx s 2 000 000 znakmi. Dĺžka šifrovania takéhoto súboru sa zvýšila oproti ostatným prípadovým úlohám na 2 sekundy. Zistili sme, že softvér Security IT už pri takomto množstve znakov potrebuje čas na zašifrovanie. Veľkosť súboru je klasicky rozdielna, s tým, že šifrovaný súbor je menší ako pôvodný. Znovu tu platí pravidlo, ktoré sme objavili aj pri

iných softvéroch, že čím je väčšia veľkosť pôvodného súboru, tým väčší rozdiel nastane medzi jeho veľkosťou a veľkosťou šifrovaného súboru. Na druhú stranu sa miesto na disku v porovnaní s predchádzajúcou bonusovou úlohou nezmenilo a taktiež platí, že šifrovaný súbor zaberá až o 44 kB menej ako pôvodný. Čas, ktorý softvér potrebuje na odšifrovanie tohto zašifrovaného súboru je 4 sekundy. Postupne zisťujeme, že softvér už v takýchto vysokých počtoch znakoch potrebuje pár sekúnd na spracovanie. No i napriek tomu, že sa tento čas zvýšil, je šifrovanie veľmi efektívne.

Našou poslednou bonusovou úlohou je šifrovanie najväčšieho dokumentu spomedzi všetkých skúmaných s počtom znakov 6 000 000. Softvér potrebuje na zašifrovanie takého súboru 4 sekundy, čo je v porovnaní ostatnými úlohami najviac. Rozdiel veľkostí pôvodného a zašifrovaného dokumentu je výrazný, pretože dosahuje až 138 555 bajtov. Znamená to teda, že Security IT zmenší šifrovaním pôvodný súbor takmer 7 násobne. Môžeme teda povedať, že pri všetkých prípadových úlohách sa veľkosť pôvodného dokumentu zmenší. Čo sa týka veľkosti, ktoré zaberajú tieto dokumenty na disku, neexistuje pravidlo, ktoré by hovorilo o tom, že šifrovaný súbor bude vždy zaberať menej miesta ako pôvodný. Konkrétne v tejto bonusovej úlohe to platí a môžeme povedať, že platí to pri všetkých dokumentoch, ktoré majú vyšší počet znakov, čiže väčšiu veľkosť. Môžeme preto povedať, že softvér zmenší veľkosť pôvodnému súboru na disku vtedy, keď dosahuje minimálne 300 000 znakov. Pri počte znakov 6 000 000 zníži túto veľkosť až o 136 kB. Čas, potrebný na odšifrovanie tohto súboru, je 4 sekundy. Zo zistených údajov sme prišli na to, že aj posledná bonusová úloha je efektívna a Security IT je veľmi výkonný šifrovací program, ktorý kvalitne a rýchlo zvláda všetky požiadavky.



Obrázok 52 Prostredie Secure IT (vlastné)

3.1.3 Prehľad výsledkov šifrovania prípadových úloh algoritmom AES

Úloha	Softvér	Čas šifrovania	Čas odšifrovania	Veľkosť/ veľkosť na disku pôvodného súboru	Veľkosť/ veľkosť na disku zašifrovaného súboru
50000 znakov	Cryptool	0 s	0 s	61 354 B/ 60 kB	61 360 B/ 60 kB
150000 znakov	Cryptool	0 s	0 s	184 076 B/ 180 kB	184 080/ 180 kB
300000 znakov	Cryptool	0 s	0 s	368 153 B/ 360 kB	368 160 B/ 360 kB
1000000 znakov	Cryptool	5 s	4 s	1 226 997 B / 1,17 MB	1 227 008 B / 1,17 MB
2000000 znakov	Cryptool	15 s	13 s	2 453 997 B / 2,34 MB	2 454 000 B / 2,34 MB
6000000 znakov	Cryptool	36 s	30 s	7 361 998 B / 7,02 MB	7 362 000 B / 7,02 MB
50000 znakov	AxCrypt	0 s	0 s	15 486 B / 16 kB	16 420 B / 20 kB
150000 znakov	AxCrypt	1 s	1 s	21 819 B / 24 kB	23 160 B/ 24 kB
300000 znakov	AxCrypt	1 s	1 s	28 560 B / 28 kB	29 163 B / 32 kB
1000000 znakov	AxCrypt	2 s	2 s	46 047 B / 48 kB	30 099 B / 32 kB
2000000 znakov	AxCrypt	2 s	2 s	64 582 B / 64 kB	22 107 B / 24 kB
6000000 znakov	AxCrypt	2 s	2 s	166 638 B / 164 kB	30 149 B / 32 kB
50000 znakov	7-Zip File Manager	1 s	1 s	15 486 B / 16 kB	12 337 B / 16 kB
150000 znakov	7-Zip File Manager	1 s	1 s	21 819 B / 24 kB	18 814 B / 20 kB
300000 znakov	7-Zip File Manager	1 s	1 s	28 560 B / 28 kB	24 738 B / 28 kB

1000000 znakov	7-Zip File Manager	1 s	1 s	46 047 B / 48 kB	25 586 B / 28 kB
2000000 znakov	7-Zip File Manager	1 s	1 s	64 582 B / 64 kB	17 382 B / 20 kB
6000000 znakov	7-Zip File Manager	2 s	2 s	166 638 B / 164 kB	22 418 B / 24 kB
50000 znakov	Secure IT	0 s	0 s	15 486 B / 16 kB	12 446 B / 16 kB
150000 znakov	Secure IT	0 s	0 s	21 819 B / 24 kB	19 067 B / 20 kB
300000 znakov	Secure IT	0 s	1 s	28 560 B / 28 kB	25 012 B / 28 kB
1000000 znakov	Secure IT	0 s	2 s	46 047 B /48 kB	26 147 B / 28 kB
2000000 znakov	Secure IT	2 s	4 s	64 582 B / 64 kB	18 584 B / 20 kB
6000000 znakov	Secure IT	4 s	4 s	166 638 B / 164 kB	28 083 B / 28 kB

3.1.4 Zhodnotenie výsledkov šifrovania prípadových úloh algoritmom AES

Pri pohľade na tabuľku prehľadu výsledkov šifrovania sme zistili, že softvér Cryptool pri prvých troch úlohách vykonáva šifrovanie za 0 sekúnd a v ďalších úlohách sa jeho rýchlosť postupne znižuje. Veľkosti jeho pôvodných a šifrovaných súborov sú však veľké v porovnaní so súbormi ostatných softvérov. Taktiež šifrované súbory zaberajú na disku rovnaký priestor ako pôvodné. Týmto môžeme povedať, že Cryptool je dobrý na šifrovanie, ale určite nie je najefektívnejší softvér pre šifrovanie prostredníctvom algoritmu AES.

Na rozdiel od Cryptoolu, softvér Axcrypt je v prvej úlohe porovnateľne rýchly, no v ďalších dvoch úlohách sa čas šifrovania zvýšil na 1 sekundu. Z tohto nám vychádza, že celkovo v prvých troch prípadových úlohách z časového hľadiska je AxCrypt menej efektívny ako Cryptool. Veľkou nevýhodou Cryptoolu je fakt, že súbory potrebné na zašifrovanie a aj výsledné šifrované súbory majú, v porovnaní s ostatnými skúmanými softvérmi, vyššiu veľkosť. Z tohto pohľadu je jednoznačne viac efektívny AxCrypt. Preto je na základe prvých troch prípadových úloh zložité vyjadriť, ktorý z týchto dvoch softvérov viac efektívny. Keď sa však pozrieme na tri bonusové úlohy, vidíme, že čas potrebný na šifrovanie, ako aj veľkosť súborov, je lepší v prospech softvéru AxCrypt a z toho môžeme povedať, že je efektívnejší.

Softvér 7-Zip File Manager je dosť podobný softvéru AxCrypt, vzhľadom na čas šifrovania aj veľkosť súborov. Keď si to bližšie priblížime, prvé tri prípadové úlohy majú rýchlosť šifrovania 1 sekundu. Tento parameter môžeme porovnať so sofvérom AxCrypt ako rovnocenný, aj napriek tomu, že v jeho prvej úlohe bola táto rýchlosť okamžitá. Na druhej strane veľkosti jednotlivých šifrovaných súborov sú v každej úlohe menšie. Keď si všimneme nasledujúce bonusové úlohy, vidíme, že rýchlosť šifrovania zostáva stále 1 sekunda, na rozdiel od bonusových úloh softvéru AxCrypt, kde rýchlosť narástla na 2 sekundy. Čo sa týka veľkosti šifrovaných súborov, tu platí, to isté, čo v prvých troch úlohách, teda, že veľkosti týchto súborov sú menšie ako pri softvéri AxCrypt. Týmto môžeme povedať, že softvér 7-Zip File Manager je efektívnejší ako AxCrypt.

Posledný softvér, ktorý sme skúmali, je Security IT. Pri porovnaní so zatiaľ najefektívnejším softvérom 7-Zip File Manager, je jeho čas šifrovania v prvých troch prípadových úlohách rýchlejšia a veľkosť súborov identická. Z týchto prvých troch úloh by bolo jasné, že efektívnejší softvér je Security IT, no pri pohľade na bonusové úlohy sa toto tvrdenie nepotvrdzuje. Čas šifrovania v druhej a tretej bonusovej úlohy je vyšší, ako je tomu pri softvéri 7-Zip File Manager a veľkosti šifrovaných súborov sú znovu rovnaké.

Pri zohľadnení všetkých doterajších výsledkov vieme povedať, že najefektívnejším softvérom pre naše základné tri prípadové úlohy je jednoznačne softvér **Security IT**, ktorého čas šifrovania a odšifrovania je minimálny a výsledný súbor získame hneď po kliknutí. Taktiež veľkosť týchto vzniknutých zašifrovaných súborov je najnižšia a zaberá na disku najmenej miesta.

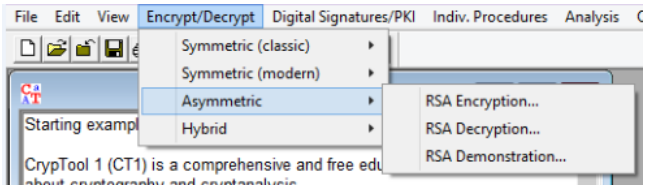
Na základe bonusových úloh sme zistili, že pri takýchto veľkých súboroch je najefektívnejší algortimus **7-Zip File Manager**, ktorý dokáže najrýchlejšie zašifrovať takéto súbory s najmenšou výslednou veľkosťou na disku.

3.1.5 Použitie šifrovacieho algoritmu RSA

Pre zopakovanie, RSA je asymetrický algoritmus, ktorý slúži na bezpečnostné služby alebo účely, umožňuje šifrovanie verejného kľúča a je používaný na zabezpečenie citlivých údajov, hlavne, keď sa odosiela cez nezabezpečenú sieť. V kryptografii RSA vie verejný aj súkromný kľúč zašifrovať správu. Na jej dešifrovanie sa použije opačný kľúč ako ten, ktorý sa použil na šifrovanie správy. Na základe tohto sa RSA stala najpoužívanejším asymetrickým algoritmom. Poskytuje metódu na zabezpečenie dôvernosti, autentickosti, integrity a nevyradenia elektronickej komunikácie a uchovávania údajov.

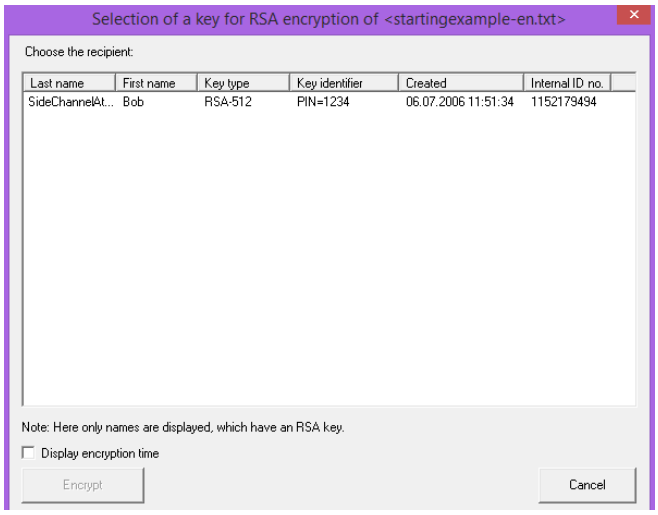
Cryptool

Program, prostredníctvom ktorého, budeme skúmať efektivitu tohto algoritmu, je softvér Cryptool. Tento softvér sme si už popisovali, tak môžme prejsť k možnosti výberu algoritmu.



Obrázok 53 Výber algoritmu RSA (vlastné)

V algoritme máme tri možnosti RSA výberu: šifrovanie, odšifrovanie a demonštrácia. Vyberieme si možnosť „RSA encryption“ a otvorí sa nám okno s nastaveniami.

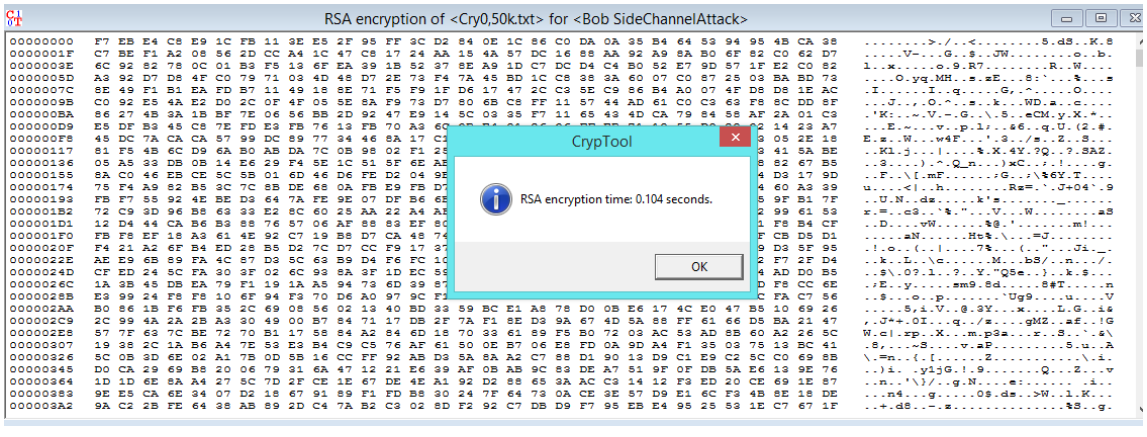


Obrázok 54 Nastavenia RSA šifrovania (vlastné)

Tu si vyberieme jedinú možnosť, ktorú nám softvér ponúkne na šifrovanie a môžeme šifrovať. Výhodou je, že si tu vieme nastaviť časovač, ktorý bude zisťovať, ako dlho prebieha šifrovanie.

Výsledok prvej prípadovej úlohy prostredníctvom algoritmu RSA softvérom Cryptool:

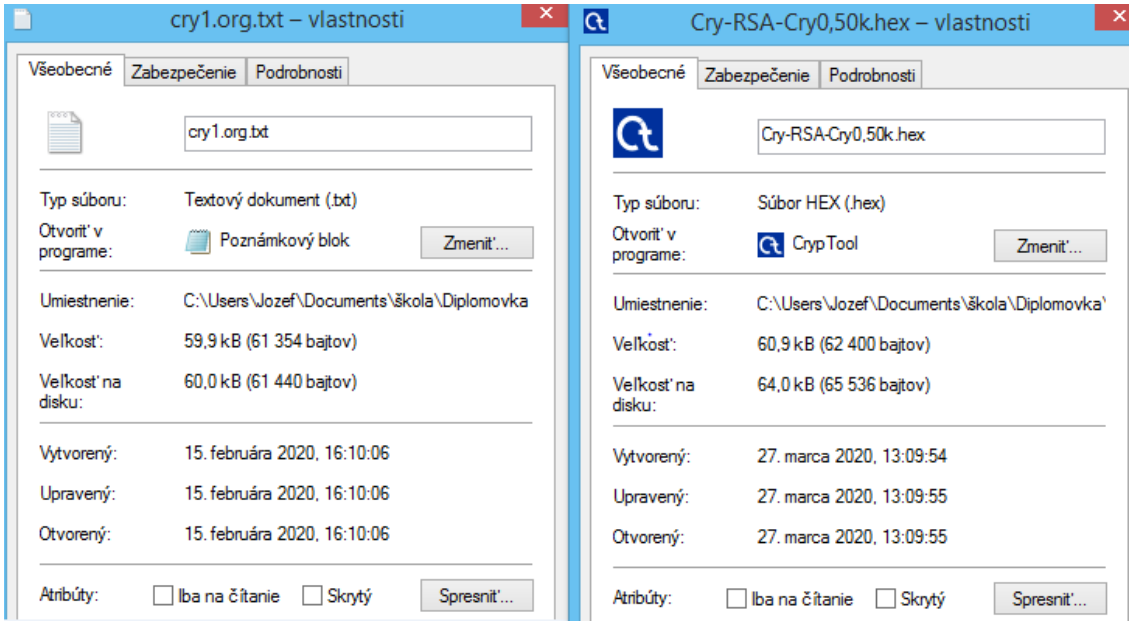
Prvou prípadovou úlohou je zašifrovať súbor s počtom znakov 50 000 prostredníctvom algoritmu RSA. Na začiatku sme si vložili tieto znaky do Cryptoolu, kde čas vkladania bol okamžitý. Po nastavení algoritmu RSA môžeme šifrovať. Výsledný zašifrovaný súbor nám vznikal 0,104 sekundy a tento čas sme získali za pomoci časovača.



Obrázok 55 zašifrovaný text (vlastné)

Veľkosť pôvodného súboru s 50 000 znakmi je 61 354 bajtov a veľkosť nového zašifrovaného súboru je 62 400 bajtov. To znamená, že po šifrovaní sa veľkosť zvýšila o 1

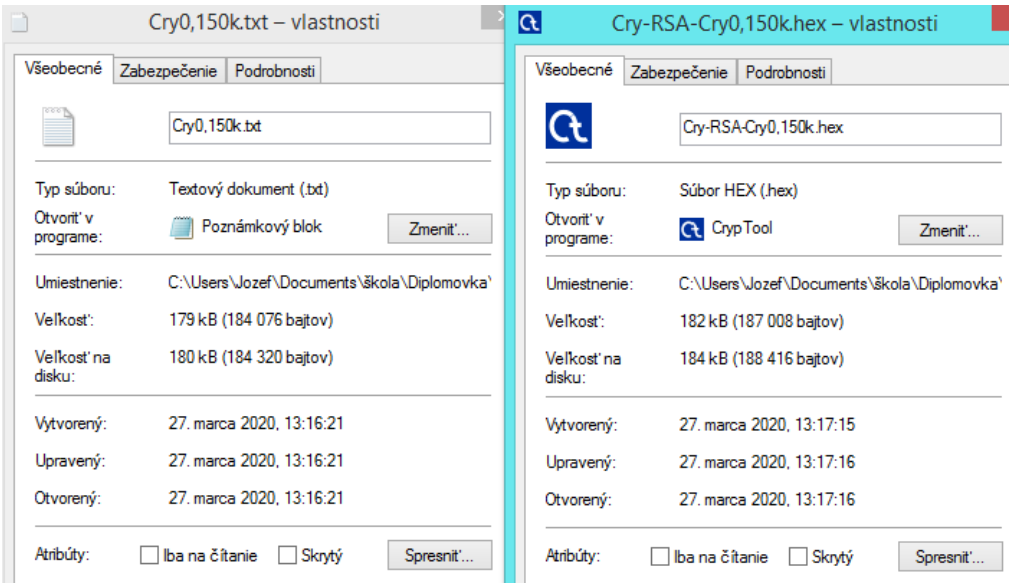
046 bajtov. Keď sa pozrieme na miesto, ktoré zaberajú tieto súbory na disku, zistíme, že pôvodný dosahuje veľkosť 60 kB a zašifrovaný o 4 kB viac, čiže 64 kB. Odšifrovanie funguje podobne ako šifrovanie, vyberieme si možnosť „RSA Decryption“ a vložíme potrebné heslo, ktoré sa nastavilo pri šifrovaní a odšifrujeme súbor. Tento proces trvá až 1,241 sekúnd. Bez hesla by sme nevedeli šifrovaný súbor odšifrovať a bol by stratený.



Obrázok 56 Vlastnoti súborov (vlastné)

Výsledok druhej prípadovej úlohy prostredníctvom algoritmu RSA softvérom Cryptool:

V druhej prípadovej úlohe budeme šifrovať súbor s 150 000 znakmi a uvidíme, aká bude zmena oproti prvej prípadovej úlohe v tomto algoritme. Znovu vložíme tento text do programu Cryptool, kde sa proces šifrovania sa vykoná okamžite. Potom vyberieme algoritmus a zašifrujeme. Čas, ktorý tento softvér potrebuje na zašifrovanie tohto algoritmu takejto veľkosti, je 0, 254 sekundy. Veľkosť pôvodného súboru je 184 076 bajtov, čo je o 2 932 bajtov ako má zašifrovaný súbor, čiže 187 008 bajtov.



Obrázok 57 Vlastnosti súborov (vlastné)

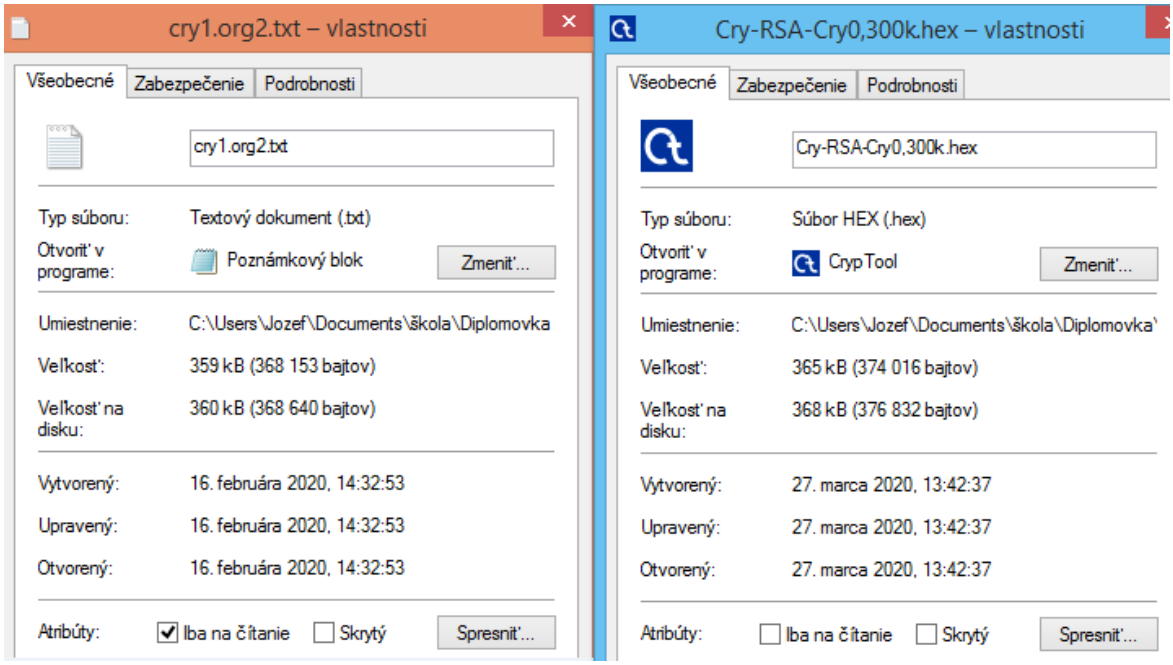
Keď sa pozrieme na veľkosť na disku, vidíme, že pôvodný súbor zaberá 180 kB a zašifrovaný dosahuje 184 kB. Z tohto teda platí, že Cryptool pri algoritme RSA zvyšuje veľkosť svojho zašifrovaného súboru.

Odšifrovanie takéhoto zašifrovaného súboru po vložení správneho hesla trvá až 3,718 sekúnd, čo je oproti času šifrovania o dosť viac.

Výsledok tretej prípadovej úlohy prostredníctvom algoritmu RSA softvérom Cryptool:

V tejto prípadovej úlohe sa budeme zaoberať šifrovaním a pozorovaním súboru s počtom znakov 300 000. Nastavíme všetky náležitosti pre šifrovania a stlačíme tlačítko „encrypt“. Následne nám Cryptool za 0,496 sekúnd zašifroval náš pôvodný súbor. Ako vidíme, rýchlosť šifrovania sa oproti predchádzajúcm dvom úlohám zvýšila len veľmi minimálne. Veľkosť pôvodného súboru je 368 153 bajtov a v porovnaní so šifrovaným súborom, ktorého veľkosť je 374 016 bajtov, vidíme, že je jeho veľkosť menšia. Pri pohľade na priestor, ktorý zaberajú jednotlivé súbory na disku, zistíme, že šifrovaný súbor s veľkosťou 368 kB ho zaberá viac ako pôvodný s 360 kB.

Keď sa pozrieme na odšifrovanie šifrovaného súboru, čas, ktorý Cryptool potrebuje na vykonanie tohto procesu, je 7,591 s.



Obrázok 58 Vlastnosti súborov (vlastné)

Výsledok bonusovej prípadovej úlohy prostredníctvom algoritmu RSA softvérom Cryptool:

Po vykonaní prvých troch úloh sme zistili, že pri všetkých je šifrovanie Cryptoolom pomerne efektívne. Preto budeme zisťovať efektívnosť šifrovania pri bonusových prípadových úlohách so súbormi s počtom znakov, podobne ako v predchádzajúcom algoritme, 1 000 000, 2 000 000 a 6 000 000.

Prvou bonusovou úlohou, ktorú budeme skúmať, je šifrovanie súboru s 1 000 000 znakov. Vyberieme si algoritmus RSA, nastavíme potrebné údaje a zašifrujeme. Čas, potrebný na zašifrovanie tohto súboru, je 1,703 sekúnd. Pôvodný súbor v tvare .txt má veľkosť 1 226 997 bajtov a zašifrovaný súbor, ktorý z neho vznikne, má 1 246 528 bajtov. Z tohto vieme povedať, že šifrovaný súbor je o 19 531 znakov väčší ako pôvodný. Z hľadiska veľkosti, ktorú zaberajú tieto súbory na disku vidíme, že pôvodný súbor dosahuje 1,17 MB a zašifrovaný 1,19 MB. V tomto prípade platí, rovnako ako pri veľkostiach, že zašifrovaný súbor zaberá väčší priestor na disku ako pôvodný.

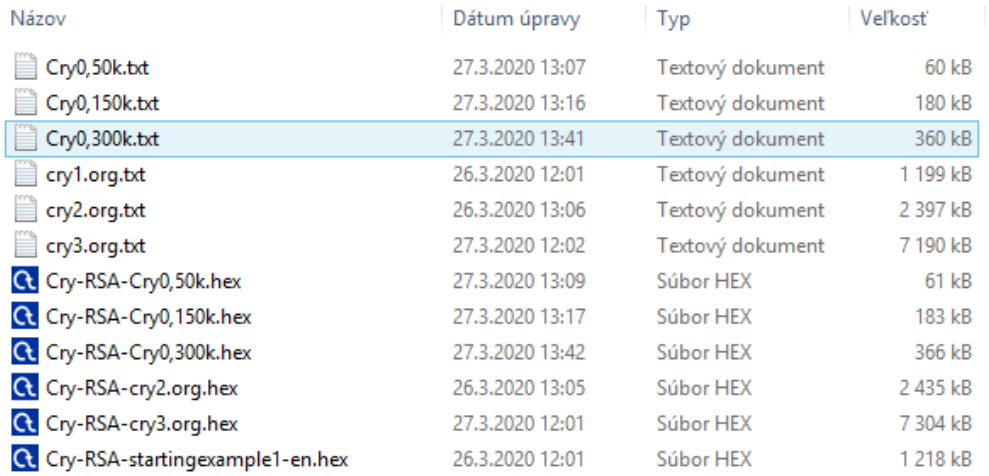
Odšifrovanie prebieha podobne ako šifrovanie, najprv vyberieme zašifrovaný súbor, potom možnosť odšifrovania, zadanie hesla a softvér začne odšifrovávať. Tento priebeh trval 17,928 sekúnd. Týmto vidíme, že odšifrovanie je oveľa náročnejšie ako šifrovanie.

V druhej bonusovej úlohe budeme šifrovať súbor s 2 000 000 znakmi. Rýchlosť, za ktorú softvér zašifruje súbor, je 3,521 sekúnd. Veľkosť pôvodného súboru je 2 453 997 bajtov a veľkosť šifrovaného dosahuje 2 492 992 bajtov. Podobne ako vo všetkých predchádzajúcich úlohách platí, že zašifrovaný súbor je väčší ako pôvodný. V tejto úlohe je tento rozdiel 38 995 bajtov. Rovnako platí aj pri zaberaní miesta na disku, že šifrovaný súbor zaberá väčší priestor ako pôvodný súbor, ktorých tieto veľkosti sú 2,37 MB a 2,34 MB.

Dĺžka odšifrovania je opäť vyššia a dosahuje až 38,436 sekúnd. Tu vidíme, že rozdiel medzi šifrovaním a odšifrovaním je pomerne veľký, pretože odšifrovanie sa vykonáva takmer 12-krát dlhšie.

Poslednou bonusovou úlohou, ktorú si priblížime, je šifrovanie súboru s počtom znakov 6 000 000 a uvidíme, akú efektivitu bude dosahovať. Po nastavení všetkých náležitostí môžeme spustiť šifrovanie. Čas, potrebný na zašifrovanie takéhoto veľkého súboru, je 10,565 sekúnd. V porovnaní s ostatnými prípadovými úlohami je to vzhľadom na počet znakov podobný výsledok. Čo sa týka veľkosti súborov, znovu vidíme, že zašifrovaný súbor je väčší. Jeho veľkosť dosahuje 7 478 912 bajtov, narozdiel od pôvodného, ktorý nadobúda veľkosť 7 361 998 bajtov. Celkovo je teda rozdiel týchto veľkostí až 117 914 bajtov. V prípade miesta, ktoré zaberajú súbory na disku, je to podobné, keďže zašifrovaný súbor zaberá 7,13 MB a pôvodný 7,02 MB. Môžeme povedať, že vo všetkých prípadových úlohách platí, že Cryptool zvýši šifrovaním veľkosť pôvodného súboru.

Odšifrovanie tohto súboru sa zvýšilo až na 145,340 sekúnd, čo je v porovnaní so šifrovaním takmer 15 krát pomalšie.



Názov	Dátum úpravy	Typ	Veľkosť
Cry0,50k.txt	27.3.2020 13:07	Textový dokument	60 kB
Cry0,150k.txt	27.3.2020 13:16	Textový dokument	180 kB
Cry0,300k.txt	27.3.2020 13:41	Textový dokument	360 kB
cry1.org.txt	26.3.2020 12:01	Textový dokument	1 199 kB
cry2.org.txt	26.3.2020 13:06	Textový dokument	2 397 kB
cry3.org.txt	27.3.2020 12:02	Textový dokument	7 190 kB
Cry-RSA-Cry0,50k.hex	27.3.2020 13:09	Súbor HEX	61 kB
Cry-RSA-Cry0,150k.hex	27.3.2020 13:17	Súbor HEX	183 kB
Cry-RSA-Cry0,300k.hex	27.3.2020 13:42	Súbor HEX	366 kB
Cry-RSA-cry2.org.hex	26.3.2020 13:05	Súbor HEX	2 435 kB
Cry-RSA-cry3.org.hex	27.3.2020 12:01	Súbor HEX	7 304 kB
Cry-RSA-startingexample1-en.hex	26.3.2020 12:01	Súbor HEX	1 218 kB

Obrázok 59 Prehľad súborov (vlastné)

3.1.6 Prehľad výsledkov šifrovania prípadových úloh algoritmom RSA softvérom Cryptool

Úloha	Softvér	Čas šifrovania	Čas odšifrovania	Veľkosť/ veľkosť na disku pôvodného súboru	Veľkosť/ veľkosť na disku zašifrovaného súboru
50 000 znakov	Cryptool	0,104 s	1,241 s	61 405 B/ 60 kB	62 400 B/ 64 kB
150 000 znakov	Cryptool	0,254 s	3,718 s	184 076 B/ 180 kB	187 008 B/ 184 kB
300 000 znakov	Cryptool	0,496 s	7,591 s	368 153 B/ 360 kB	374 016 B/ 368 kB
1 000 000 znakov	Cryptool	1,703 s	17,928 s	1 226 997 B/ 1,17 MB	1 246 528 B/ 1,19 MB
2 000 000 znakov	Cryptool	3,521 s	38,436 s	2 453 997 B/ 2,34 MB	2 492 992 B/ 2,37 MB
6 000 000 znakov	Cryptool	10,656 s	145,340 s	7 361 998 B/ 7,02 MB	7 478 912 B/ 7,13 MB

3.1.7 Prehľad výsledkov šifrovania prípadových úloh algoritmom AES sofvtérom Cryptool

Úloha	Softvér	Čas šifrovania	Čas odšifrovania	Veľkosť/ veľkosť na disku pôvodného súboru	Veľkosť/ veľkosť na disku zašifrovaného súboru
50000 znakov	Cryptool	0 s	0 s	61 354 B/ 60 kB	31 360 B/ 60 kB
150000 znakov	Cryptool	0 s	0 s	184 076 B/ 180 kB	184 080/ 180 kB
300000 znakov	Cryptool	0 s	0 s	368 153 B/ 360 kB	368 160 B/ 360 kB
1000000 znakov	Cryptool	5 s	4 s	1 226 997 B / 1,17 MB	1 227 008 B / 1,17 MB
2000000 znakov	Cryptool	15 s	13 s	2 453 997 B / 2,34 MB	2 454 000 B / 2,34 MB
6000000 znakov	Cryptool	36 s	30 s	7 361 998 B / 7,02 MB	7 362 000 B / 7,02 MB

3.1.7 Zhodnotenie a porovnanie výsledkov šifrovania prípadových úloh algoritmom AES a RSA prostredníctvom softvéru Cryptool

Po zistení výsledkov, ktoré nám vznikli zo šifrovania týmito dvoma algoritmami prostredníctvom Cryptoolu, sme zistili, že na základe troch prípadových úloh je z hľadiska času efektívnejšie šifrovanie algoritmom AES. Čo sa týka veľkostí, vidíme, že v AES sú jednotlivé veľkosti súborov takmer identické a miesto, ktoré zaberajú na disku je rovnaké. Z tohto vychádza, že Cryptool pri šifrovaní algoritmom AES nezväčšuje veľkosť pôvodného súboru na disku. Na druhej strane, keď sa pozrieme na šifrovanie algoritmom RSA, vidíme, že šifrovaním sa táto veľkosť zväčší. Ďalším rozdielom medzi šifrovaním týmito dvomi algoritmami je rýchlosť odšifrovania. Cryptool zvládne zašifrovaný súbor algoritmom AES odšifrovať okamžite, narozdiel čoho odšifrovaváníe algoritmu RSA mu trvá dlhšie. Na základe všetkých zohľadnených vlastností a zistených poznatkov môžeme povedať, že v softvéri Cryptool v troch prípadových úlohách je efektívnejšie používať algoritmus **AES**.

V našich bonusových úlohách nastala zmena, pretože čas, potrebný na zašifrovanie sa pri algoritme AES zvýšil a dosahuje vyššie hodnoty ako pri algoritme RSA. Nič to však nemení na tom, že čo sa týka veľkosti súborov a času odšifrovania je efektívnejší a tým pádom naďalej platí, že v softvéri Cryptool je efektívnejšie šifrovať algoritmom **AES** ako RSA.

Záver

Zabezpečenie súkromných údajov na počítači je v súčasnosti jeden z veľkých problémov, s ktorými sa bežní ľudia alebo firmy stretávajú v každodennej praxi. Používanie kryptografických softvérov, ktoré využívajú najmodernejšie kryptografické algoritmy, sa stala bežnou súčasťou života väčšiny ľudí a firiem, ktorí si chcú chrániť svoje údaje pred virtuálnymi útočníkmi. Existuje mnoho takýchto softvérov, ktoré slúžia na šifrovanie rôznych údajov, súborov a taktiež konkrétnych častí disku alebo celého pevného disku.

V diplomovej práci sme si ako cieľ vytýčili porovnanie niektorých šifrovacích softvérov pri šifrovaní rovnakých prípadových úloh, pričom sme porovnávali aj rozdielne šifrovacie algoritmy v rámci jedného softvéru. V teoretickej rovine sme si popísali kryptografiu, rozdelenie šifrovacích algoritmov a následne sme si každý algoritmus vysvetlili. Tiež sme sa zamerali aj na popis najlepších šifrovacích softvérov podľa hodnotenia spoločnosti HeimdalSecurity. Priblíženie šifrovacích algoritmov a šifrovacích softvérov bolo dôležité pre naplnenie čiastkových cieľov, na základe ktorých sme vybrali vhodné softvéry. Uvedené vybrané softvéry sme následne v praktickej časti tejto diplomovej práce porovnávali. Väčšina, na rozdiel od softvéru Cryptool, umožňuje šifrovať len v šifrovacom algoritme AES, čo neumožnilo porovnávanie viacerých šifier vo viacerých softvéroch.

Ako sme v úvode spomínali, tretia kapitola bola venovaná praktickej časti, kde sme ako výber použili softvéry Cryptool, AxCrypt, 7-Zip File Manager, Secure IT. V týchto softvéroch sme popísali spôsob zašifrovania troch hlavných prípadových úloh s počtom znakov 50 000, 150 000, 300 000 a troch bonusových prípadových úloh s vyšším počtom znakov 1 000 000, 2 000 000 a 6 000 000. Zisťovali sme nasledovné parametre šifrovania, ktoré sme potom zapisovali do výslednej tabuľky: čas šifrovania, celková veľkosť pôvodného súboru, celková veľkosť zašifrovaného súboru, veľkosť na disku pôvodného súboru, veľkosť na disku zašifrovaného súboru a čas odšifrovania. Šifrovanie každej prípadovej úlohy prebehlo vo všetkých vybraných softvéroch prostredníctvom algoritmu AES. Na základe týchto parametrov sa porovnávala efektívnosť jednotlivých softvérov. Ďalšou otázkou, ktorou sme sa zaoberali, bolo ozrejmiť problematiku, ktorý šifrovací algoritmus spomedzi AES a RSA je v softvéri Cryptool efektívnejší v šifrovaní našich prípadových úloh.

Po praktickom šifrovaní vo všetkých šifrovacích softvéroch sme porovnávali jednotlivé parametre. Dozvedeli sme sa, že najmenej efektívnym softvérom, spomedzi vybraných, prostredníctvom algoritmu AES, je Cryptool. Nasledujú ho softvéry Axcrypt a 7-Zip File Manager, ktoré sú z pohľadu porovnávania veľmi podobné. Môžeme potvrdiť, že najefektívnejším šifrovacím softvérom sa spomedzi prvých troch prípadových úloh stal Security IT, ktorého parametre boli najefektívnejšie vzhľadom na čas šifrovania, čas odšifrovania a aj veľkostí jednotlivých súborov. Pozoruhodným sa stalo to, že pri výsledkoch šifrovania bonusových úloh, ktoré sa skladali z vyššieho počtu znakov nad 1 000 000, sa poradie efektívnosti softvérov zmenilo. Najefektívnejším softvérom sa pre takýto počet znakov stal 7-Zip File Manager. Ďalším zisťovaním, ktoré sme skúmali, bolo porovnať šifrovanie prípadových úloh algoritmom AES a algoritmom RSA v softvéri Cryptool. Na základe rovnakých parametrov sme porovnali ich hodnoty a vyzistili sme, že algoritmus AES je efektívnejší na šifrovanie v tomto softvéri ako RSA.

Môžeme teda skonštatovať, že hlavný cieľ, ktorým bolo zistiť najefektívnejší softvér na šifrovanie prípadových úloh, má dve riešenia. Prvé riešenie je pre tri prípadové úlohy s menším počtom znakov, kde je najefektívnejší softvér Security IT, a druhé riešenie pre prípadové úlohy s vyšším počtom znakov, kde sa najefektívnejším stal 7-Zip File Manager. Ďalej sme dospeli k záveru, že najefektívnejším algoritmom pre šifrovanie v softvéri Cryptool je AES.

Prínos týchto zistených výsledkov je v tom, že sme sa skúmaním presvedčili, ktorý z týchto softvérov podľa veľkostí súborov je najefektívnejší na zabezpečenie svojich súkromných údajov pred nebezpečnými hackermi. Dôležitým faktom je, že nie vždy, to čo sa javí ako najlepšie pre menšie súbory, je vhodné aj pri veľkých súboroch.

Zoznam použitej literatúry

[1] *Cryptography* [elektronický zdroj].[2018], online. [cit. 2020-3-20]. Dostupné na: <https://www.techopedia.com/definition/1770/cryptography>

[2] *Šifrovanie informácií* [elektronický zdroj], online. [cit. 2020-3-20]. Dostupné na: <https://gkmke.sk/informatika/4.rocnik/SifrovanieInformacii.pdf?fbclid=IwAR2GY-m29d1RpWKCgesC8PyN4NLjSRS3t4iUt9VW_VztdsoDveDe3grLe2k>

[3] *Cryptography* [elektronický zdroj].[2018], online. [cit. 2020-3-20]. Dostupné na: <https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking>

[4] SMIRNOFF, Peter - M. TURNER, Dawn. *Symmetric Key Encryption - why, where and how it's used in banking* [elektronický zdroj].[2019], online. [cit. 2019-3-20]. Dostupné na: <https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking?fbclid=IwAR3sEv3T-xM3UNDIz-TQabMXY8I-GEqkKGPK193DZSV4SZ4VX7fMFhUaz74>

[5] SHUBHAMUPADHYAY. *Data encryption standard (DES) | Set 1* [elektronický zdroj]., online. [cit. 2019-3-20]. Dostupné na: <https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/>

[6] HENRY, Jasmine. *3DES is Officially Being Retired* [elektronický zdroj].[2018], online. [cit. 2019-3-20]. Dostupné na: <https://www.cryptomathic.com/news-events/blog/3des-is-officially-being-retired>

[7] *AES* [elektronický zdroj]., online. [cit. 2020-3-20]. Dostupné na: <http://www.kryptografie.wz.cz/data/aes.html>

[8] TAYLOR, Shelby. *Kompletní návod na standard pokročilého šifrování (AES)* [elektronický zdroj].[2018], online. [cit. 2019-3-20]. Dostupné na: <https://cs.wizcase.com/blog/kompletni-navod-na-standard-pokrocileho-sifrovani-aes/>

[9] *RC2* [elektronický zdroj].[2018], online. [cit. 2019-3-20]. Dostupné na: <https://en.wikipedia.org/wiki/RC2>

[10] *Introduction to Blowfish* [elektronický zdroj].[2014], online. [cit. 2019-3-20]. Dostupné na: <https://www.splashdata.com/splashid/blowfish.htm>

[11] *Camellia (cipher)* [elektronický zdroj]., online. [cit. 2019-3-21]. Dostupné na:

<https://cryptography.fandom.com/wiki/Camellia_(cipher)>

[12] *CRYPTOMUSEUM. One-Time Pad* [elektronický zdroj].[2015], online. [cit. 2019-3-21]. Dostupné na: <https://www.cryptomuseum.com/crypto/otp/index.htm>

[13] *RC4* [elektronický zdroj].[2020], online. [cit. 2019-3-21]. Dostupné na:

<https://en.wikipedia.org/wiki/RC4>

[14] *Salsa20* [elektronický zdroj]., online. [cit. 2019-3-21]. Dostupné na:

<http://www.crypto-it.net/eng/symmetric/salsa20.html?tab>

[15] ROUSE, Margaret – BRUSH, Kate – ROSENCRANCE, Linda – COBB, Michael. *asymmetric cryptography (public key cryptography)* [elektronický zdroj].[2020], online. [cit. 2019-3-21]. Dostupné na:

<https://searchsecurity.techtarget.com/definition/asymmetric-cryptography>

[16] *Symmetric vs. Asymmetric Encryption – What are differences?* [elektronický zdroj]., online. [cit. 2019-3-21]. Dostupné na: <https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences>

[17] SamanvayaPanda. *ElGamal Encryption Algorithm* [elektronický zdroj]., online. [cit. 2019-3-21]. Dostupné na: <https://www.geeksforgeeks.org/elgamal-encryption-algorithm/>

[18] ROUSE, Margaret – COBB, Michael – HAZAN, Fred – RUNDATZ, Frank. *RSA algorithm (Rivest-Shamir-Adleman)* [elektronický zdroj].[2018], online. [cit. 2019-3-21]. Dostupné na: <https://searchsecurity.techtarget.com/definition/RSA>

[19] *Digital Signature Algorithm* [elektronický zdroj].[2020], online. [cit. 2019-3-21]. Dostupné na: <https://en.wikipedia.org/wiki/Digital_Signature_Algorithm>

[20] *Diffie-Hellman, RSA, DSA, ECC and ECDSA – Asymmetric Key Algorithms* [elektronický zdroj]., online. [cit. 2019-3-21]. Dostupné na: <https://www.ssl2buy.com/wiki/diffie-hellman-rsa-dsa-ecc-and-ecdsa-asymmetric-key-algorithms>

[21] ROUSE, Margaret – BURR, Jonathan. *elliptical curve cryptography (ECC)* [elektronický zdroj].[2005], online. [cit. 2019-3-22]. Dostupné na:

<https://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography>

[22] RANADEEPIKA2409. *What are Hash Functions and How to choose a good Hash Function?* [elektronický zdroj]., online. [cit. 2019-3-22]. Dostupné na:

<https://www.geeksforgeeks.org/what-are-hash-functions-and-how-to-choose-a-good-hash-function/>

[23] RIJNETU, IOANA. *The most Popular Free Encryption Software Tools to Protect Your Data* [elektronický zdroj].[2019], online. [cit. 2019-3-22]. Dostupné na:

<https://heimdalsecurity.com/blog/free-encryption-software-tools/>

[24] *VeraCrypt* [elektronický zdroj]., online. [cit. 2019-3-22]. Dostupné na:

<https://www.veracrypt.fr/en/Home.html>

[25] VIGO, Jesus. *Apple's FileVault 2 encryption program: A cheat sheet* [elektronický zdroj].[2018], online. [cit. 2019-3-22]. Dostupné na:

<https://www.techrepublic.com/article/apples-filevault-2-encryption-program-a-cheat-sheet/>

[26] *DiskCryptor* [elektronický zdroj].[2020], online. [cit. 2019-3-22]. Dostupné na: <https://en.wikipedia.org/wiki/DiskCryptor>

[27] *7-Zip* [elektronický zdroj].[2020], online. [cit. 2019-3-22]. Dostupné na:

<https://en.wikipedia.org/wiki/7-Zip>

[28] *Axcrypt* [elektronický zdroj].[2020], online. [cit. 2019-3-22]. Dostupné na:

<https://www.axcrypt.net/>

[29] TOWNSEND SECURITY. *AES vs. DES Encryption: Why Advanced Encryption Standard (AES) has replaced DES, 3DES and TDEA* [elektronický zdroj].[2018], online. [cit. 2019-3-23]. Dostupné na: <https://blog.syncsort.com/2018/08/data-security/aes-vs-des-encryption-standard-3des-tdea/>

[30] CIHODARIU, Miriam. *The Best Encrypted Messaging Apps You Should Use Today* [elektronický zdroj].[2019], online. [cit. 2019-3-23]. Dostupné na: <https://heimdalsecurity.com/blog/the-best-encrypted-messaging-apps/?fbclid=IwAR3BlWT2Y5s6bUJ9YA8_0gr2lfrFHTF6_OFQEjeq3jR5TP7yh3sVvnhiRUU>

[31] JOHNSTON, Nicole. *Secure IT 200 Review* [elektronický zdroj].[2018], online. [cit. 2019-3-23]. Dostupné na: <https://www.toptenreviews.com/encryption-software-secure-it-review>

[32] *Feistel cipher* [elektronický zdroj].[2020], online. [cit. 2019-3-23]. Dostupné na: <https://en.wikipedia.org/wiki/Feistel_cipher>

[33] Redakce PCT. *Moderní metody šifrování* [elektronický zdroj].[2005], online. [cit. 2019-3-23]. Dostupné na: <https://pctuning.tyden.cz/software/ochrana-soukromi/4711-moderni_metody_sifrovani>

[34] LYNN, Ben. *Cyclic Groups* [elektronický zdroj].[2005], online. [cit. 2019-3-23]. Dostupné na: <https://crypto.stanford.edu/pbc/notes/group/cyclic.html>

[35] *torproject* [elektronický zdroj].[2020], online. [cit. 2019-3-23]. Dostupné na: <https://www.torproject.org/download/>