Testy odolnosti kryptografických metod

Bc. Adam Volf

Diplomová práce
2018

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

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

Jméno a příjmení:	**Bc. Adam Volf**
Osobní číslo:	**A16217**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Učitelství informatiky pro střední školy**
Forma studia:	**prezenční**
Téma práce:	**Testy odolnosti kryptografických metod**
Téma anglicky:	**Resistance Tests Based on Cryptographic Methods**

Zásady pro vypracování:

1. **Popište druhy kryptografických metod, jejich působnost v komerční sféře a moderní standardy.**
2. **Proveďte výběr metod a aplikací pro jejich praktické otestování.**
3. **Specifikujte HW platformu pro testování.**
4. **Otestujte zvolené metody kryptografie na vzorcích databáze.**
5. **Zhodnoťte výsledky testu kryptografických metod.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **PAAR, Christof a Jan. PELZL. Understanding cryptography: a textbook for students and practitioners. New York: Springer, c2010. ISBN 978-3-642-04101-3.**
2. **CURTIN, Matt. Brute force: cracking the data encryption standard. New York: Copernicus Books, c2005. ISBN 0-387-20109-2.**
3. **EDITED BY G. SOMASUNDARAM a Alok SHRIVASTAVA. Information storage and management: storing, managing, and protecting digital information. Indianapolis, IN: Wiley Publishing, 2009. ISBN 9780470294215.**
4. **SANCHEZ, Ignacio, Apostolos MALATRAS a Iwen COISEL. A security analysis of email communications. Luxembourg: Publications Office of the European Union, 2015. ISBN 978-9 2-7 9-6 6503-5.**
5. **BONEH, Dan a Victor SHOUP. A Graduate Course in Applied Cryptography [online]. Stanford, CA, 2017 [cit. 2017-11-29]. Dostupné z: https://crypto.stanford.edu/ dabo/cryptobook/BonehShoup_0_4.pdf**

Vedoucí diplomové práce: **Ing. David Malaník, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **1. prosince 2017**

Termín odevzdání diplomové práce: **16. května 2018**

Ve Zlíně dne 11. prosince 2017

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
garant oboru

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně, dne .....

.....
podpis diplomanta

ABSTRAKT

Kryptografie se využívá k utajení informací už od starověku a v moderní době je nepostradatelnou součástí procesu ochrany dat. S narůstající výpočetní silou klesá doba nutná k provedení kryptoanalytických útoků na zašifrovaná data. Tato práce popisuje kryptografické metody a jejich standardy aktuálního softwaru pro ochranu šifrováním dat a na základě testů zhotovených pomocí speciálního hardwaru a kryptoanalytických metod hodnotí jejich odolnost.

Klíčová slova: Kryptografie, odolnost, šifry, kryptoanalýza, šifrování, bezpečnost, kryptografické metody, standardy.

ABSTRACT

Cryptography is used to conceal information since antiquity and in modern age they are indispensable part of data protection. With growing computing power, the time needed to complete cryptoanalytic attack against encrypted data decreases. This work describes cryptographic methods and their standards of current software for data encryption and evaluates their resistence based on tests, made with special hardware and cryptoanalytic methods.

Keywords: Cryptography, resistence, ciphers, cryptoanalysis, encryption, safety, cryptographic methods, standards.

Rád bych poděkoval vedoucímu mé diplomové práce, panu Ing. Davidu Malaníkovi Ph.D. za odborné vedení a trpělivost při konzultacích. Také bych chtěl poděkovat za poskytnutí výpočetní techniky pro účely spojené s cíli diplomové práce. Dále bych chtěl poděkovat rodině a přátelům, kteří mě podporovali během mého studia.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

Bezpečnost uskladněných dat je každým dnem důležitější a velmi často se můžeme doslechnout o úniku částí databází (Mall, LinkeIn) [76]. Avšak i po odcizení těchto dat slouží k jejich ochraně (i mnoha uživatelů) kryptografické metody, pomocí kterých jsou tyto data zašifrovány.

Kryptografické metody jsou v moderní informatice používány již desítky let a v průběhu této doby dochází k technologickému vývoji nejen těchto metod a výkonu výpočetní techniky, ale i k způsobům, jak zabezpečení těchto metod prolomit.

Šifrovací algoritmy, jako je AES, Twofish, Serpent a další, jsou pouze jádrem tohoto zabezpečení a pokud jsou implementovány nesprávným způsobem, je velmi pravděpodobné, že tuto ochranu lze pomocí detailní znalosti problému obejít. Stejně tak, jak je tomu obecně u nástrojů jakéhokoliv pracovníka, je důležité, aby věděl jak s těmito nástroji pracovat a jak je použít. K tomuto účelu slouží návody, normy a zavedené způsoby k jejich operaci. Podobně je tomu u šifrovacích algoritmů, které vyžadují znalost v oblasti jejich operace a implementace do větších systémových celků. K tomuto slouží standardy, které vydávají důvěryhodné autority (NIST, ISO/IEC) zabývající se jejich popisem, evaluací, aktualizací některých těchto metod a i jejich případnému zrušení platnosti. Tyto standardizované metody jsou základem pro konstrukci šifrovacího softwaru, jenž je využíván ve vládních i nevládních organizacích chránící citlivé data svých uživatelů, klíčových strategií, ale i přísně tajného materiálu, který bývá často odtajněn až po desítkách let.

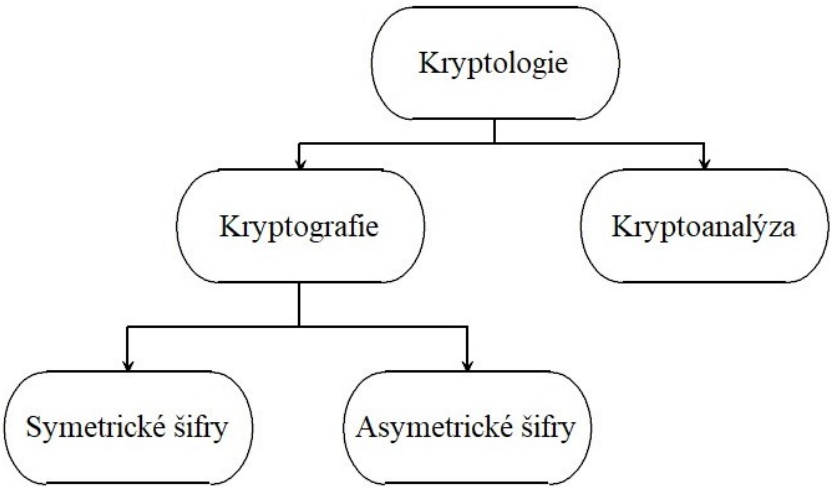
Při výběru zabezpečení dat pomocí šifrovacího softwaru, je proto důležité najít informace o použitých kryptografických metodách, jejich standardech a také jejich analýze, jenž dokáže přiblížit, na jak dlouho budou tyto metody bezpečné, vzhledem k rychlosti vývoje výpočetní techniky (Moorův zákon).

Známou kryptoanalytickou metodou pro ověření bezpečnosti těchto kryptografických metod je útok hrubou silou a jeho varianty, jejichž cílem je hledání klíčů nebo hesel za co nejkratší možnou dobu, která je závislá na výpočetním výkonu, použité kryptografické metodě, rychlosti hledacího algoritmu a také síle uživatelského hesla, jenž bývá často nejslabším článkem tohoto zabezpečení.

I. TEORETICKÁ ČÁST

1 KRYPTOGRAFICKÉ METODY

Pod slovem kryptografie se nám často jako první může vybavit asociace: šifrování emailů, zabezpečení přístupu k webům, internetové bankovnictví nebo i známé zařízení z doby druhé světové války Enigma. Kryptografie je v dnešní době velmi blízce spojena s informatikou a moderní elektronickou komunikací, avšak první zmínky jejího využití se objevují už 2000 let před naším letopočtem a to při utajování informací pomocí nestandardních, tajných hieroglyfů v starověkém Egyptu. [1]



Obrázek 1 Diagram popisující souvislost věd kryptologie a kryptografie.[2]

Kryptografie je věda, která se zabývá metodami a technikami utajování významu zpráv a řadí se spolu s kryptoanalýzou pod obecnější vědní obor kryptologii. [3]

1.1 Symetrická kryptografie

Symetrická kryptografie nebo také kryptografie s tajným klíčem, využívá jediný klíč pro obě operace: šifrování i odšifrování. Pokud se tato metoda využívá při komunikaci, musí se účastníci domluvit na tajném klíči, kterým budou tyto tajné zprávy šifrovat. Symetrické algoritmy se dále kategorizují do dvou skupin: Proudové šifry a blokové šifry. Proudová šifra operuje většinou v rozmezí 1 bitu, bytu informace ve stejný čas zatím, co bloková šifra operuje na blocích dat ve stejný čas a to nejčastěji o délce 128 bitů. [4] [44]

1.1.1 AES (Rijndael)

AES nebo také pokročilý šifrovací standard (dokument NIST FIPS PUB 197 [26]) je mezinárodně uznaná symetrická šifrovací metoda, jejímž známým uživatelem je především

vláda Spojených Států, která jím nahradila zastaralý DES. Ačkoliv termín standard v jeho názvu pouze ukazuje na jeho aplikaci vlády Spojených Států - je také součástí několika komerčních standardů, kterými jsou například Internetový bezpečnostní standard IPsec, TLS, Wi-Fi šifrovací standard IEEE802.11i, bezpečnostní protokol SSH (Secure Shell), Internetový telefon Skype a mnoho bezpečnostních produktů po celém světě. Tato bloková šifra je založena na algoritmu Rijndael, jenž byla výhercem ve velmi důkladném výběrovém procesu, jako nástupce DES s délkou bloku 128 bitů, která používá několik cyklů šifrování v závislosti na délce klíče, typicky 10 - až 14 krát. Klíče jsou běžně generovány samotným softwarem, ale užívají se i uživatelské hesla, což ze zřejmých důvodů není doporučováno. Jak bylo už výše řečeno, AES šifra je založena na Rijandael, ale délka bloku ve standardu je vždy pouze 128 bitů na rozdíl od Rijandaelu, jehož variace podporují výběr mezi 128, 192 a 256 bity. Schéma algoritmu AES se skládá z 3 různých vrstev obsahující kryptografické operace. Každá z těchto vrstev transformuje všech 128 bitů dat. Každý cyklus, s výjimkou prvního, se skládá ze všech 3 vrstev:[5] [2]

- Klíčová vrstva, transformuje data na základě klíče.
- Substituční vrstva, nelineární transformace použitím tabulek.
- Difuzní vrstva, difuze stavu bitů. [6]

1.1.2 Kuznyechik

Tato bloková šifra, také známa pod GOST34.12-2015 byla vyvinuta společným úsilím Centra Bezpečnosti Informací a Speciální komunikace Federální Bezpečnostní Služby Ruské Federace s účastí společnosti Informační technologie a komunikační systémy. Šifrovací algoritmus byl schválen a představen Federální agenturou v červnu 2015. Stejně jak šifra AES, Kuznyechik je blokovou šifrou využívající substitučně-permutační sítě k zašifrování 128 bitových bloků dat. Šifra využívá 256 bitového klíče k vytvoření deseti 128 bitových klíčů, které algoritmus využívá v cyklech šifrování. V každém cyklu se využívá funkce: Nelineární, bijektivní mapování bytů, lineární transformace a vrstvy XOR pro prolnutí cyklových klíčů se zašifrovaným textem. [13] [14]

1.1.3 CAST5

CAST5 nebo CAST-128 je šifrovací algoritmus, jenž náleží do skupiny využívající Feistelovu síť. Zajímavá je především tím, že je navržena, aby odolala Bihamově rotační, na klíči založené kryptoanalýze. Od ostatních šifer kandidátů na AES se liší velikostí datového

bloku o 64 bitech. Klíče využívá o velikosti 128 bitů, avšak existuje mladší verze CAST6, která podporuje 256 bitové klíče. CAST5 algoritmus poskytuje 12 nebo 16 cyklů šifrování pomocí Feistelovy sítě. Využívá rotační funkci k obraně proti lineárním a diferenčním útokům. [15] [16]

1.1.4 Twofish

Twofish symetrická bloková šifra byla publikována v roce 1998 a byla jedním z pěti finalistů kandidátů na AES šifru. Není patentována a je zdarma ke stažení pro všechny uživatele. Stejně jak je tomu u AES, je velikost bloku 128 bitů s velikostí klíče od 128 do 256 bitů. Parametry šifry jsou stejně tak, jako u AES požadavky výběrového procesu vládního orgánu NIST, jenž měl za úkol nahradit zastaralou šifru DES. Twofish nejprve rozdělí blok dat na 4 části po 32 bitech, tyto slova projdou „bílením“ XORem, které se nachází i na konci algoritmu. První 2 části bloku jsou vstupem pro funkci g, která je složena z S-boxů a MDS maticí, již následuje PHT transformace. Druhou část bloku následuje to samé, avšak v následujícím kole se prohodí. Tento cyklus se opakuje 16 krát. Ačkoliv byl Twofish navrhnut s výkonem jako prioritou, je uváděno, že současný AES je rychlejší. Twofish je velmi modulární a je možné ho modifikovat, aby byl rychlejší na úkor bezpečnostních vrstev. Tato šifra našla využití v aplikacích a extenzích pro emailovou, textovou komunikaci, šifrování a kompresi souborů, je také součástí nabídky symetrických šifer standardu PGP. [6] [7]

1.1.5 Serpent

Serpent je jako ostatní AES kandidáti 128 bitová symetrická bloková šifra, která využívá 32 cyklů permutací a substitucí. Ačkoliv měl ze všech 3 předchozích šifer nejkomplexnější zabezpečení, skončil ve výběrovém procesu jako druhý z finalistů. Protože hlavním faktorem NIST soutěže bylo zabezpečení, rozhodl se tým Serpentu pro návrh co nejbezpečnější šifry. Při návrhu tedy počítali s tím, že bude třeba, aby nový standard AES měl co nejdelší životnost. I když měl Serpent v kategorii hardwarového výkonu nejlepší výsledky, skončil v soutěži jako druhý kvůli rozdílu v rychlosti. [8] [9]

1.1.6 Blowfish

Blowfish je symetrická bloková šifra, kterou můžeme efektivně využít při šifrování a zabezpečení dat. Tvůrcem šifry je Bruce Sheiner a navrhl ji jako alternativu k tehdejším populárním symetrickým šifrám v roce 1993. Poprvé byla prezentována v Cambridgském

workshopu šifrových algoritmů. Od samotného počátku měla být šifra úplně zdarma, nepatentovaná, nelicencovaná a bez autorských práv – alternativa k DES. Od jejího počátku, byla šifra důkladně analyzována a začalo se ji dostávat přijetí jako silného šifrovacího algoritmu. První implementací Blowfishe bylo v softwaru LabView. Šifrovací algoritmus Blowfishe šifruje bloky o 64 bitech. Základní funkce, které jsou využívány v algoritmu, byly vybrány s hlavní prioritou v rychlosti. Algoritmus užívá Fiestelovu síť a můžeme ji rozdělit do následujících částí: [11] [12]

1. Klíčová expanze – převedení klíče do matic pomocí S-Boxů.
2. Šifrování dat – průchod Feistelovou sítí, jejíž každý cyklus se skládá z klíče-závislé permutace a substituce.
3. Odšifrování dat – je stejné jako šifrování, ale jednotlivé funkce se provádí se v opačném pořadí. [11] [12]

Blowfish našel uplatnění v následujícím softwaru: Advanced CS – šifrování a zálohování souborů. Access Manager – manažer pro správu hesel systému Windows. AEdit: Textový procesor s podporou šifrování. [12]

1.1.7 IDEA

Idea je bloková šifra, kterou navrhli Xuejia Lai a James L. Masey a jenž byla poprvé představena v roce 1991. Je založena na šifře PES (Proposed Encryption Standard), originálně byla IDEA nazývána IPES, kde I stojí za Improved. Jedním z důvodů vzniků této šifry byla zastaralá šifra DES, kterou měla nová šifra nahradit. Jedním ze zajímavých aspektů této šifry je, že se vyhýbá použití jakýchkoliv převodních tabulek nebo S-Box funkcí. V době kdy se pracovalo na známém PGP emailovém a souborovém šifrovacím produktu (Phil Zimmermann), byla to právě šifra IDEA, jenž byla jejich první volbou. Algoritmus blokové šifry pracuje s 64 bitovými bloky a 128 bitovým klíčem. Základem tohoto algoritmu spočívá ve využití operací ze tří různých algebraických skupin. Pro generaci klíče slouží následující postup: Protože všechny použité algebraické funkce operují na základě 16 bitových čísel, jsou textové bloky rozděleny do čtyř 16 bitových pod-bloků. Další procedura produkuje šest 16 bitových klíčových pod-bloků pro každý šifrovací cyklus ze 128 bitového klíče. Pro dokončení výstupní transformace je třeba 52 klíčových pod-bloků, proto ještě je třeba vygenerovat poslední čtyři. Výsledkem je 52, 16 bitových klíčových pod-bloků. [17] [18]

1.1.8 Camellia

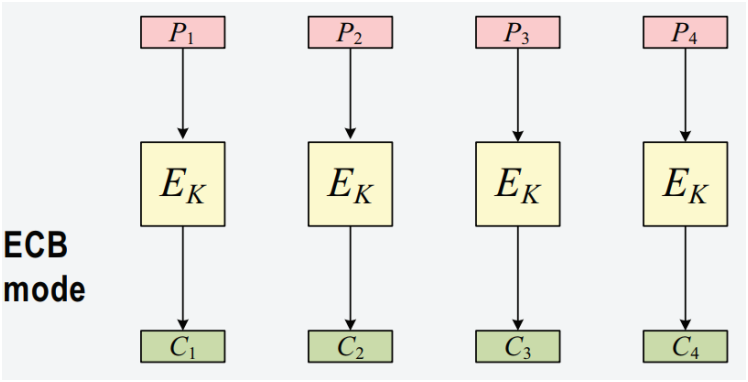
Tato šifra byla vyvinuta společným úsilím společností Nippon Telegraph, korporací Telephone a Mitsubishi Electric v roce 2000. Camellia je blokovou šifrou o délce bloku 128 bitů, která podporuje délky klíčů 128, 192 a 256 bitů a je charakterizována svou využitelností pro softwarové i hardwarové implementace a také svou vysokou úrovní bezpečnosti, kterou poskytuje. Z praktického hlediska je navržena tak, aby umožnila flexibilitu v softwarových i hardwarových implementacích široce užívaných přes Internet a v mnoha dalších aplikacích. Struktura šifry představuje klasickou Feistelovu síť, ale obsahuje speciální operace, které jsou vloženy po každých šesti kolech této sítě. Tyto speciální operace mají pozitivní dopad v komplikování některých druhů útoků na šifru, ale na druhou stranu ničí klasickou Feistelovu strukturu, což má malé negativní důsledky na šifru. Algoritmus šifry má 18 až 24 kol kryptografických operací, které zahrnují aplikaci S-boxů, které mají skvělé bezpečnostní vlastnosti proti diferenční a lineární kryptoanalýze. [59] [60]

1.1.9 Operační módy blokových šifer

Existuje několik odlišných způsobů implementace blokových šifer, které se nazývají operačními módy. Populárními módy jsou zejména CBC a novější XTS, který byl zaveden spolu s příchodem Windows 10 (build 1511) v softwaru Microsoft Bitlocker jenž je součástí operačního systému Windows. [3]

1.1.9.1 ECB – Electronic Codebook

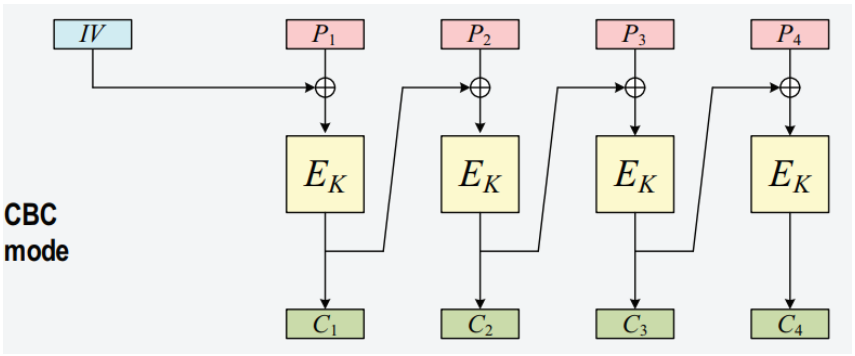
Jeden z nejstarších druhů implementace blokové šifry, původně určený pro algoritmus DES. Mód má jisté fundamentální, obecně známé slabiny a to především v tom, že výstupní šifrovaná data nejsou dostatečně náhodná. Tento mód proto není doporučován a nehodí se pro bezpečné šifrování dat. Mód implementace šifry je zobrazen na obrázku číslo 1. Pod textem. P_i je vstupní nešifrovaný text. Označení E_k na obrázku je pro samotnou blokovou šifru o délce K. C_i je výstupní blok dat zašifrovaný šifrou E_k.[46]



Obrázek 2 Diagram popisující operační mód ECB. [46]

1.1.9.2 CBC,CFB a OFB

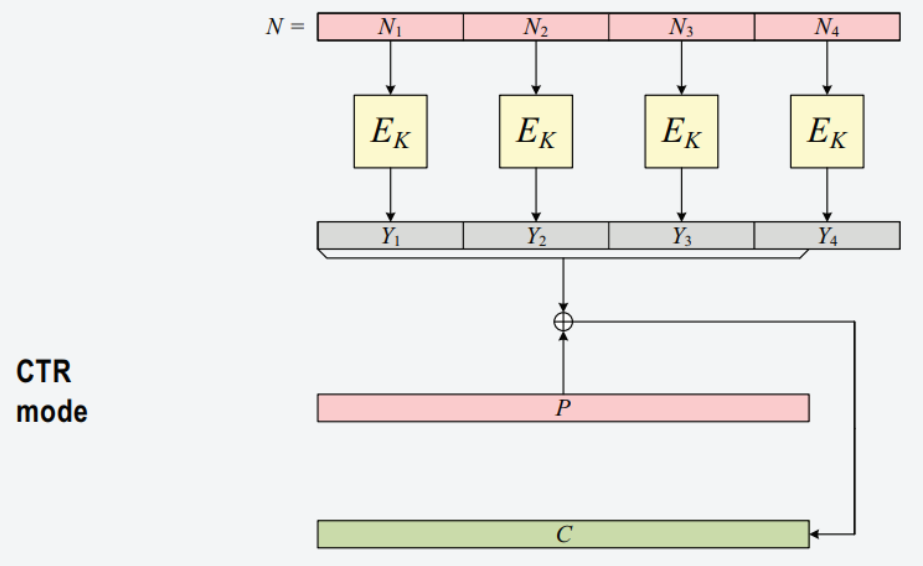
CBC (Cipher Block Chaining), CFB (Cipher Feedback) a OFB (Output Feedback) jsou tři dnes již klasické módy využívající (IV) inicializační vektor. Tyto módy byly původně popsány ve standardu FIPS 81 a sdílejí některé základní technické rysy, všechny jsou relativně bezpečné, pokud uživatel využívá náhodného inicializačního vektoru. Jednou z největších slabin těchto módu je, že lze modifikovat šifrovaný text, který bude do jisté míry podobný původnímu. Z těchto tří módů je nejvíce používaný CBC, jež je zobrazen na obrázku 2 pod textem. Na obrázku je inicializační vektor zobrazen blokem IV. V obrázku můžeme vidět spojení textových bloků pomocí funkce XOR s předchozími šifrovanými bloky do řetězu – odtud pochází název tohoto módu.[46]



Obrázek 3 Diagram popisující operační mód CBC. [46]

1.1.9.3 CTR

Tento mód byl poprvé doporučen Diffe a Hellmanem ve stejném období jako předchozí 3 módy, avšak nebyl obsažen v první várce FIPS doporučení. Vládní organizace NIST mód přidala k ostatním doporučeným schématům až v roce 2001. V porovnání s ostatními módy je jednoduší a disponuje lepší výkonnostní charakteristikou, která není na úkor bezpečnosti. Navíc operační mód podporuje šifrováním bloků v náhodném pořadí, což je výhodou například u šifrování pevných disků. Na obrázku schématu módu CTR je oproti předešlým módům použita hodnota nonce pod označením N_{1-4} a výstupy blokové šifry Y_{1-4}, které jsou spojeny pomocí funkce XOR se vstupním textem P, což vytváří zašifrovaný text C. [46] [47]

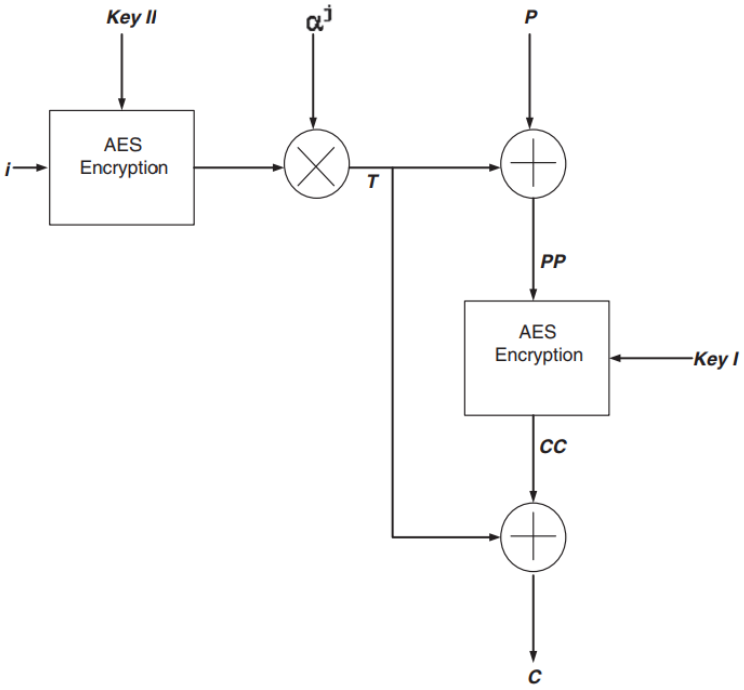


Obrázek 4 Diagram popisující operační mód CTR. [46]

1.1.9.4 XTS

XTS je jeden z novějších módů, jehož oblastí využití je šifrování dat na uložišti a není doporučen pro jakékoliv jiné využití (je nevhodný pro komunikaci přes síť). Tento mód nabízí výkonové zlepšení oproti tradičním schématům a je již využíván v několika disk-šifrujících aplikacích, jako je například Bitlocker a VeraCrypt. Hlavní odlišnosti od předchozích módů je použití dvou klíčů a bloková šifra, která je tady použita dvakrát. Na obrázku 4 je zobrazen mód šifrování XTS s šifrou AES. Vstupní hodnota i je zde hodnota „tweak“, která označuje číslo (umístění) bloku dat, tato hodnota se společně s daty zašifru-

je pomocí logických operací součinu a XOR, jak je znázorněno na obrázku pod textem.[46]

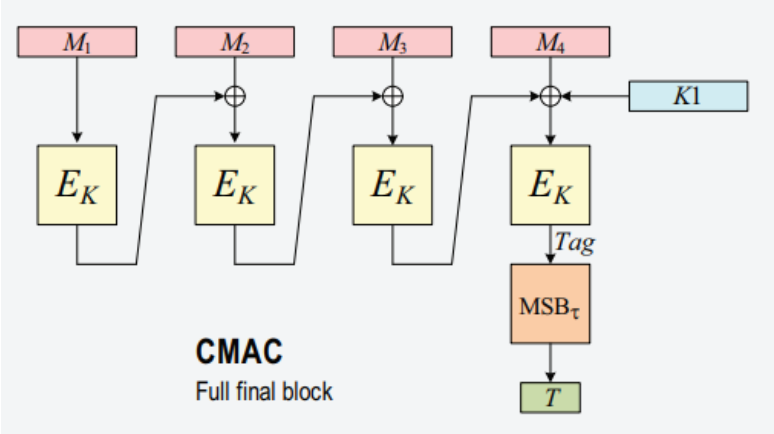


Obrázek 5 Diagram popisující operační mód XTS. [69]

1.1.10 Autentizované operační módy blokových šifer a hašovacích funkcí

1.1.10.1 CMAC

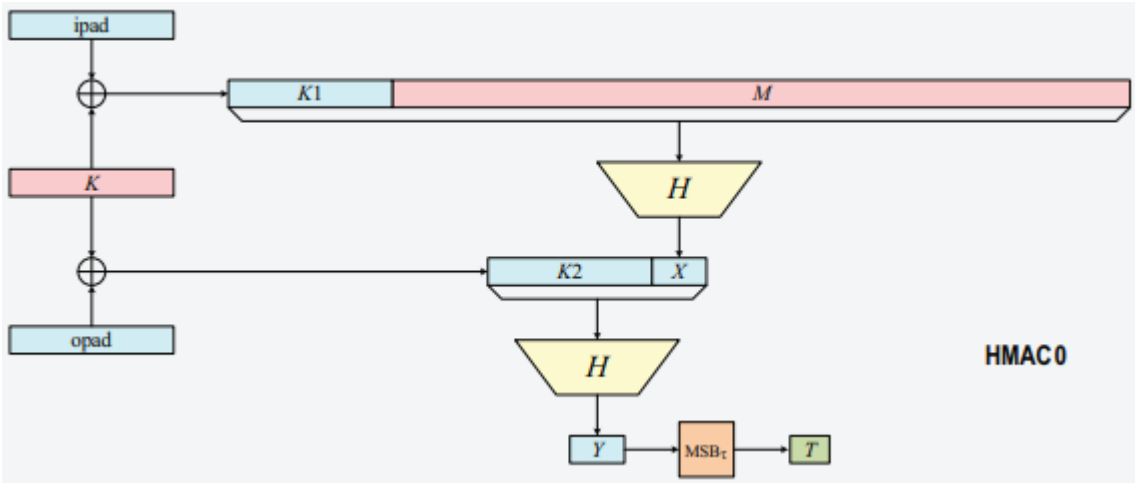
Operační mód blokových šifer CMAC patří do skupiny využívající autentizaci jako (HMAC, GMAC). Tento mód je běžně využíván se silnou 128 bitovou blokovou šifrou a poskytuje ověřitelné bezpečnostní vlastnosti bez vážných závad. CMAC je především mód navržený k detekci záměrné, neautorizované modifikaci dat a také nechtěných náhodných změn. V doporučení od NIST SP800-38b[48] je spojována především se standardy DES, AES a TDEA. V obrázku 5 můžeme vidět schéma módu CMAC, který na zprávy M_{1-4} aplikuje blokovou šifru (AES). Výstupem módu s šifrou a příslušným klíčem je autentizovaný kód T na základě zprávy M.[46] [48]



Obrázek 6 Diagram popisující operační mód CMAC. [46]

1.1.10.2 HMAC

HMAC operační mód je především využíván pro hašovací funkce a nachází využití i jako pseudo-náhodná funkce. Předností je zde jednoduchost a silná konstrukce, která vedla k široké standardizaci (nejen v NIST standardech). Mód HMAC je nejvíce využíván při autentizaci komunikace, při níž odesílatel zprávy vytvoří hodnotu MAC, která je formulována za pomocí tajného klíče a samotné zprávy. Na obrázku 6 je schéma autentizačního operačního módu HMAC využívající hašovací funkci H. Mód je doporučen využívat se standardizovanou, iterovanou hašovací funkcí (jako jsou hašovací funkce rodiny MD a SHA) v dokumentu FIPS 198-1[49] a RFC 2104.[46] [49]

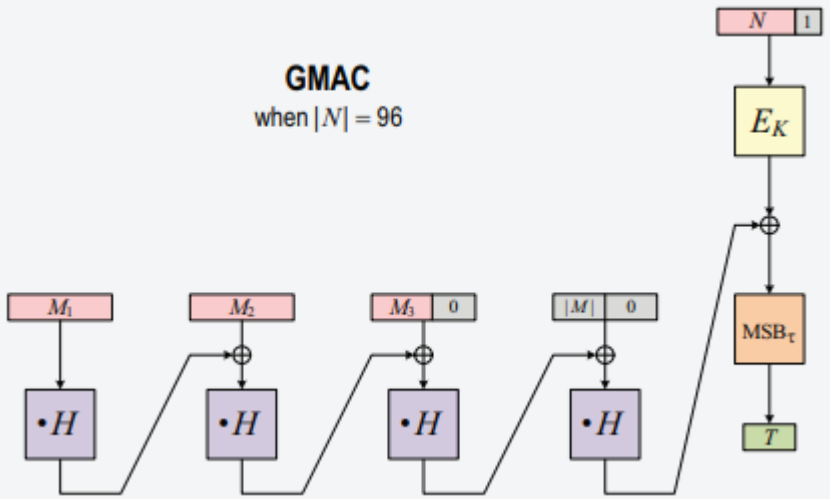


Obrázek 7 Diagram popisující operační mód HMAC. [46]

1.1.10.3 GMAC(GCM)

Tento mód se výrazně liší od všech ostatních autentizačních módů a to protože jako jediný využívá ve svém algoritmu hodnotu nonce. GMAC je speciální verzí módu GCM (Galois

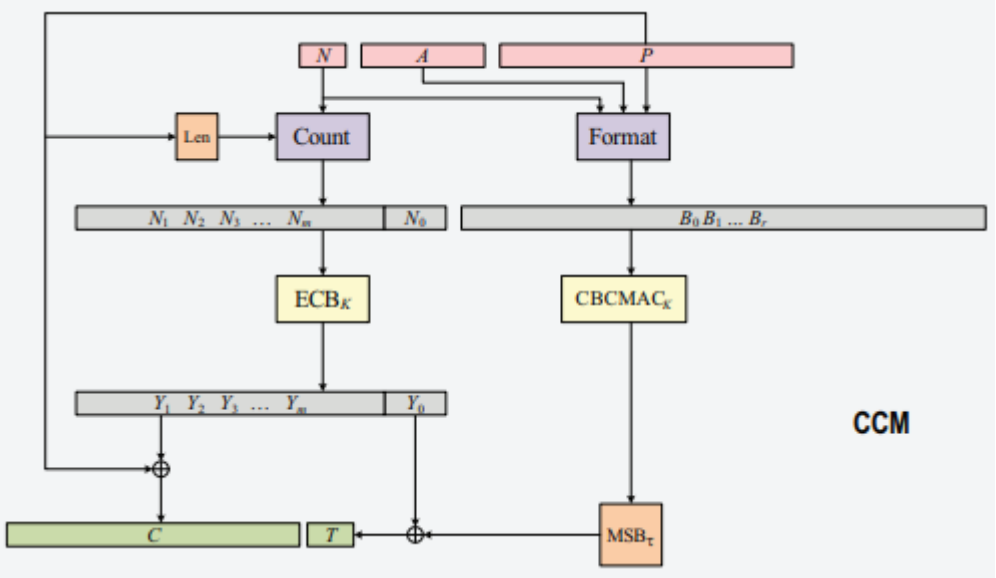
Counter Mode). Mezi jeho kladné vlastnosti patří především skvělý výkon, díky možné paralelizaci a je široce využíván, také kvůli tomu, že nemá žádné závažné bezpečnostní nedostatky. GCM poskytuje také silnou autentizaci dat a dokáže detekovat náhodné modifikace i účelné, neautorizované modifikace. Mód GMAC v obrázku 7 využívá spolu s hašovací funkcí i šifrovací algoritmus (AES) specifikovaný dokumentací NIST, jehož vstupem je hodnota nonce N.[46] [50]



Obrázek 8 Diagram popisující operační mód GMAC. [46]

1.1.10.4 CCM

CCM se řadí do skupiny operačních módu s autentizací dat a zároveň poskytuje jistotu jejich bezpečného utajení. Mód je založen na prověřených blokových šifrách o délce bloku 128 bitů, jako je například AES (doporučení NIST SP 800-38C [51]). Schéma bylo vyvinuto autory Whitting, housley a Ferguson a je kombinací CTR šifrování s CBC-MAC ověřením. Využití tohoto módu je především v paketovém prostředí a není navržen pro použití jako proudový algoritmus. Vstupem CCM jsou 3 základní elementy: Data nachystané pro autentizaci a šifrování (nazývané náklad); Související data – hlavička, která bude zašifrována, ale neautentizována; Unikátní hodnota „nonce“, přiřazena k nákladu i hlavičce. Na obrázku 8 je znázorněno schéma mód CCM se vstupními hodnotami nonce N, hlavičky A a nešifrovaným textem P. Výstupem tohoto algoritmu je zašifrovaný text C s autentizačním kódem T. [46][51]



Obrázek 9 Diagram popisující operační mód CCM. [46]

1.2 Asymetrické kryptografie

Asymetrická kryptografie také známa pod názvem kryptografie s veřejným klíčem je kryptografickým schématem, který vyžaduje dva odlišné klíče, jedním z nichž je tajný klíč a druhý je veřejný. Ačkoliv jsou tyto klíče odlišné, jsou matematicky propojeny. Termín asymetrický klíč pochází tedy z použití dvou nestejných klíčů.[43]

1.2.1 RSA

RSA, pojmenován svých tvůrců Rivest, Shamir, Adleman, je šifrovací algoritmus na bázi systému veřejného klíče. Systém byl navržen, aby nahradil zastaralý standard NBS. Inspirací tvůrcovského týmu byly práce publikované Diffie a Hellmanem několik let zpátky, kteří se myšlenkou takovéhoto šifrovacího systému již zabývali.[10] [19]

Základy této šifry stojí na dvou hlavních myšlenkách: [10]

1. Šifrování na bázi veřejného klíče. Tento základní blok umožňuje zprávu opatřit dalším zabezpečením, před jejím odesláním. V této šifře jsou dva druhy klíčů. Veřejný klíč, který slouží pouze k zašifrování a klíč k odšifrování, který je tajný a vlastní ho pouze příjemce zprávy. Tento tajný klíč musí být vygenerován takovým způsobem, aby nešel odvodit z veřejného klíče. [10]
2. Digitální podpis. Někdy je třeba si ověřit totožnost odesílatele zprávy (pomocí podpisu) a k tomu slouží odesílatelův dešifrovací klíč, podpis může být později ověřen

kýmkoliv použitím odpovídajícího veřejného klíče. Proto nelze podpis zfalšovat – odesílatel nemůže popřít, že je tvůrcem zprávy. [10]

Samotné šifrování je založeno na složitosti součinu **n** dvou náhodných, vysokých prvočísel **p** a **q**. I když je tento součin veřejný, nedokážeme z něj odvodit prvočísla. Vzorec šifrování a dešifrování vypadá následovně: [10] [19]

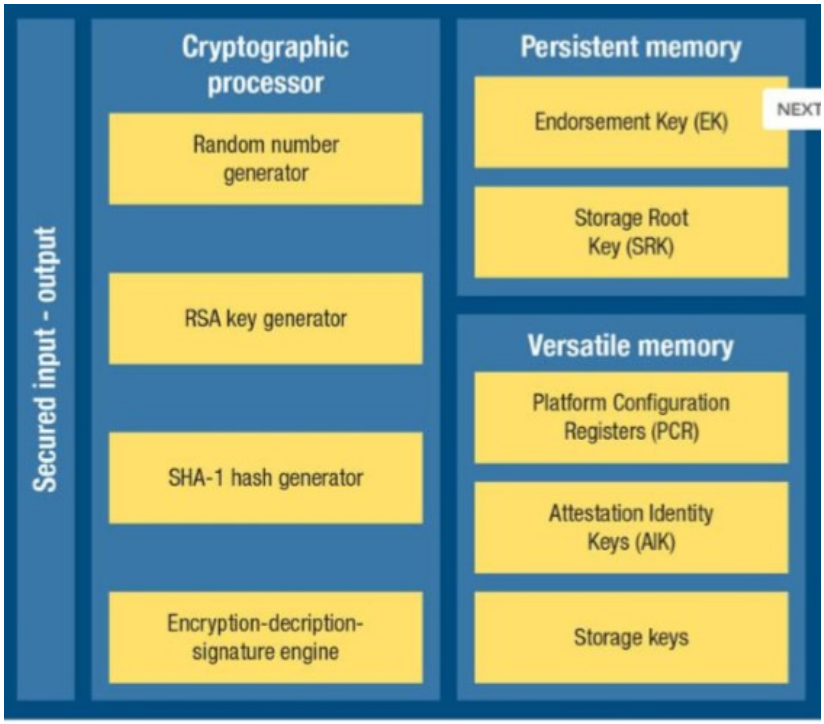
$$\mathrm{M} \equiv \mathrm{D}(\mathrm{C}) \equiv C^d \pmod{\mathrm{n}} \quad (1)$$

$$\mathrm{C} \equiv \mathrm{E}(\mathrm{M}) \equiv M^e \pmod{\mathrm{n}} \quad (2)$$

[19]

1.2.1.1 TPM (Trusted platform module)

Tento čip využívající RSA klíčový generátor slouží k ověření platnosti přístupu uživatele k datům zařízení, ve kterém je nainstalován. Je popsán mezinárodním standardem, jenž detailně specifikuje verzi 1.2 a novější 2.0, jeho využití při ukládání šifrovacích klíčů. TPM bývá často součástí základních desek osobních počítačů nebo notebooků, ale jde zakoupit i samostatně. Dalším využitím TPM čipu je například v detekci hardwaru, softwaru a firmwaru uživatelova počítače. Na obrázku 9 jsou zobrazeny moduly, které TPM čip používá k zabezpečení dat uživatele. [70]



Obrázek 10 Kryptografické moduly TPM čipu.[70]

1.2.2 Diffe-Hellman

Asymetrické schéma Diffe-Hellman bylo poprvé představeno svými tvůrci (po nichž je pojmenováno) v roce 1976. Je to specifická výměna kryptografických klíčů, jenž umožňuje dvěma stranám, které o sobě nemají žádnou předešlou znalost, navázat sdílené tajné klíče přes nezabezpečený komunikační kanál. Tento klíč může být potom použit k zašifrování následující komunikace pomocí symetrických šifer. Algoritmus závisí na náhodném výběru velkých čísel a prvočísel, pomocí kterých si každá strana vypočítá svůj tajný klíč pro tuto komunikaci. [43]

Výměna klíčů vypadá následovně: Uživatelé A a B se dohodnou na dvou velkých prvočíslech n a g. Tyto prvočísla nemusí být tajné.

1. Uživatel A si vybere další velké náhodné číslo x a vypočítá c pomocí rovnice:

$$c = g^x mod\ n \tag{3}$$

   Číslo c odešle uživateli B.

2. B nezávisle vybere další náhodné velké číslo y a vypočítá d.

$$d = g^y mod\ n \tag{4}$$

   Číslo d odešle uživateli A.

3. A vypočítá tajný klíč K_1 pomocí rovnice:

$$K1 = d^x mod\ n \tag{5}$$

4. B vypočítá tajný klíč K_2 pomocí rovnice:

$$K2 = c^y mod\ n \tag{6}$$

[43]

1.3 Hašovací funkce

Hašovací funkce slouží jako obtisk dat. Tento obtisk je menší, než jsou původní data a měl by být pro každé nestejné data odlišný (bezkolizní). Kryptografické hašovací funkce jsou nezbytně důležité pro autentizační protokoly a digitální podpisy. Velké využití se pro ně našlo zejména při ukládání hesel a to díky vlastnosti, že nelze zrekonstruovat původní hodnotu z otisku hesla vytvořeného hašovací funkcí [45]

1.3.1 Rodina funkcí Merkle-Damgard

Funkce Merkle-Damgard pojmenována po svých tvůrcích R. Merkle a I. Damgard v roce 1989, je jednou z nejšíře používanou hašovací funkcí, proto je také mnoho novějších funkcí inspirováno právě podle MD. Konstrukce MD hašovací funkce probíhá v několika krocích: Zpráva M je nejprve rozdělena do vstupních bloků K o jednotné velikosti a každý blok je kompresován funkcí, pokud je zpráva příliš krátká, tak se přidává odsazení. Použitím komprese zajišťuje hašovací funkce bez-koliznost. [20]

1.3.2 Rodina funkcí Secure Hash Algorithm

SHA nebo Secure Hash Algorithm je považován za kryptograficky bezpečnou formu ukládání dat u které není běžné rekonstruování původního stavu i za pomoci velkého výpočetního výkonu. Vláda Spojených Státu Amerických standardizovala více než 6 SHA algoritmů s nejstaršími verzemi SHA-0 a SHA-1 v 90. letech minulého století. Funkce SHA-0 byla kvůli nedostatkům brzo nahrazena funkcí SHA-1, jejímž vylepšením je výpočetní krok navíc, který adresuje nedostatky SHA-0. V následujícím desetiletí se stala novějším přírůstkem série SHA-2, ve které se nacházejí variace SHA-224, SHA-256, SHA-384 a SHA-512. Tato série byla navržena, tak aby generovala ze dvou různých dokumentů - dva různé sety hodnot hašovací funkce, tedy aby byla bezkolizní. Tyto funkce byly publikovány mezi lety 2001-2004 a jsou robustnější než jejich předchůdci. Pracují s délky bloků 224 až 1024 bitů. Jejich původ pochází od vládní organizace NSA. [20]

1.3.3 Whirlpool

Whirlpool je hašovací funkce založena na blokové šifře AES, vytvářející hašovací kód o 512 bitech pro vstupní zprávu o maximální délce menší než 2^{256} bitů. Využívá 512 bitový klíč a pracuje na 512 bitových blocích dat. Tato hašovací funkce byla vytvořena Vincentem Rijmenem, který je také spolutvůrce Rijndaelu. Whirlpool je jedna ze dvou hašovacích funkcí schválená evropským projektem NESSIE, jenž se snaží prosadit skupinu silných kryptografických technologií různého typu, včetně blokových, symetrických šifer a hašovacích funkcí. [21]

1.3.4 Streebog

Streebog byl uveden v roce 2010 a jeho výstupem je 512 nebo 256 bitová hašovací hodnota a maximální délkou vstupní zprávy je 2^{512}. Kompresní funkcí Streebogu je přes 12 cyklů šifry podobné AES. Streebog oficiálně nahrazuje přechozí standard GOST 34.11-94, který byl teoreticky prolomen. Hlavním rozdílem mezi Streebogem a jeho předchůdcem je v jeho kompresní funkci, která je založena na blokové šifře se substitučně-permutační síti s blokovou a klíčovou délkou rovné 512 bitům. Místem nasazení této hašovací funkce je především v oblasti digitálních podpisů ve spolupráci s asymetrickou kryptografií, jak je uvedeno v předchozím standardu GOST R 34.10-2012.[22] [23]

1.3.5 RIPEMD-160

Tato hašovací funkce je postavena na základu 256 bitové MD4, která byla představena v roce 1990 Ronem Rivestem. Jak napovídá název, výstupem této funkce je 160 bitová hašová hodnota. Společně s SHA-1, rychlejší verzí RIPEMD-128 byla funkce RIPEMD-160 uvedena v mezinárodním standardu v roce 1997. Hlavním místem využití této funkce je v poskytování digitálního otisku před použitím algoritmu pro digitální podpis. U předchozí verze, RIPEMD, bylo v roce 1997 dokázáno, že je možné, aby došlo ke kolizím. Tato událost zapříčinila vylepšení hašovací funkce, jejímž výsledkem byly RIPEMD-128 a RIPEMD-160.[71]

2 KRYPTOGRAFICKÉ STANDARDY

Kryptografické standardy popisují ověřené metody, funkce, algoritmy a způsoby zabezpečení dat, jejichž zdrojem jsou často vládní organizace, které navíc mohou ověřovat způsob jejich implementaci ve vládních i nevládních subjektech a udělují jim certifikaci po důsledném prověření a analýze implementace použité kryptografické metody. [75]

2.1.1 FIPS PUB 197

Standard popisuje AES algoritmus využívaný k zabezpečení elektronických dat. Agenturou pro správu tohoto standardu je Laboratoř informačních technologií náležící Národnímu institutu standardů a technologie (NIST) USA. Dále popisuje oblast aplikace a to zejména pro ochranu utajených informací na úřadech a agenturách USA. Také doporučuje využívání tohoto standardu v nevládních organizacích. NIST sleduje vývoj analýzy algoritmu a jak je tomu i u ostatních standardů, bude i jeho kvalifikace přezkoumána každých 5 let. V případě objevení nových průlomů v technologii nebo matematické slabosti algoritmu bude pro NIST důvodem k re-evaluaci standardu a k poskytnutí nezbytných revizí. [26]

2.1.2 Secure Hash Standard FIPS 180-4

Tento standard popisuje hašovací algoritmy SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 a SHA-512/256 určené pro výpočet kondenzované reprezentace elektronických dat. Velikost zpracované zprávy se pohybuje od 160 do 512 bitů v záležitosti na zvoleném algoritmu. Algoritmy standardu SHS jsou typicky využívané v kombinaci s dalšími šifrovacími algoritmy. Stejně jak standard FIPS PUB 197 i tento standard je využíván k utajování citlivých informací federálních úřadů, agentur USA i nevládních organizací. [25]

2.1.3 Derivační funkce PBKDF

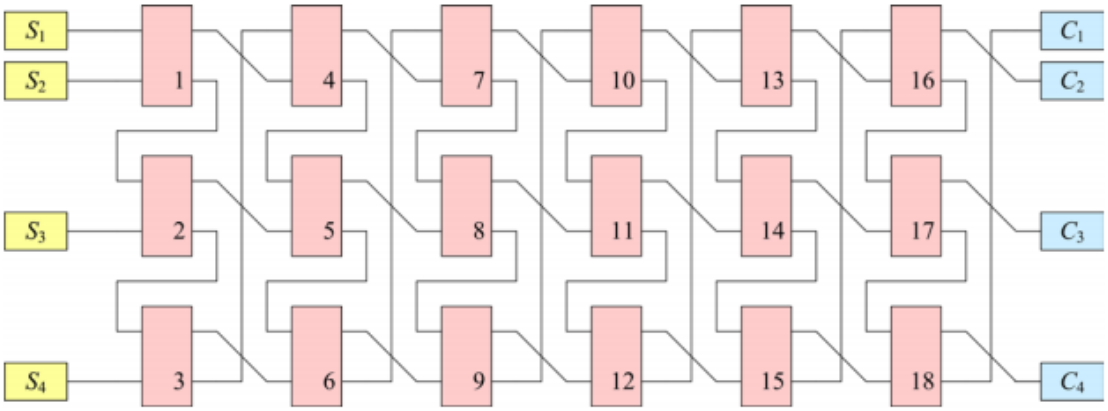
PBKDF vytváří derivovaný klíč ze základního klíče a dalších parametrů. V hesle-založené klíč-derivující funkci je základním klíčem heslo a dalšími parametry je zde sůl a počet iterací. V dokumentu doporučení od NIST jsou specifikovány dvě verze PBKDF1 a PBKDF2, kde starší verze je určena pouze pro současné aplikace z důvodu kompatibility. Druhá verze je doporučena pro všechny nové aplikace. Typickým příkladem posloupnosti operací k využití této funkce je uváděna v těchto krocích: [53] [54]

1. Vyber sůl S a počet iterací c.

2. Vyber délky oktet pro derivační klíč dkLen.
3. Aplikuj klíč-derivační funkci na heslo, sůl a počet iterací k vytvoření derivovaného klíče.
4. Použij derivovaný klíč z výstupu funkce. [53] [54]

2.1.4 Klíč balící algoritmus (Key Wrap)

Doporučení od vládní organizace NIST detailně popisuje deterministickou, autentizovanou šifrovací metodu blokové šifry AES nazývanou Key Wrap (Klíčový obal). Tato metoda je navržena, aby zabezpečila důvěrnost a nefalšovanost či integritu kryptografických klíčů. Algoritmus této metody zabezpečení zaručuje, že každý výstupní bit má netriviální závislost na každém výstupním bitu i když jsou vstupní data větší než je velikost jednoho vstupního bloku. Této charakteristiky je docíleno na úkor propustnosti v porovnání s ostatními metodami zabezpečení. Metodu je doporučeno používat v případě, že je délka chráněného klíče delší než délka bloku použité blokové šifry nebo pokud je hodnota zabezpečených dat příliš vysoká. [62][63]



Obrázek 11 Diagram popisující klíč balící algoritmus Key Wrap. [63]

Hlavním komponentem Key Wrapu je použitá bloková šifra. Klíč této metody je nazýván KEK (Klíč šifrující klíč). Doporučená délka bloku šifry je 128 bitů, jako je algoritmus AES s délkou klíče 128, 192 a 256 bitů. Délka KEK přímo ovlivňuje bezpečnost algoritmů proti útoku hrubou silou. Obrázek 10 nad textem popisuje funkci obalového algoritmu aplikovanou na 4 vstupních pod-blocích (S_1-S_4). Každá linka nese pouze jeden pod-blok a každý z 18 růžových čtverců představuje průchod šifrou spolu s klíčem KEK. Na pravé straně jsou zobrazeny výstupní bloky (C_1 až C_4). [62][63]

2.1.5 GOST 34.12-2015

Tento standard Ruské Federace definuje základní blokové šifry používané jako kryptografické metody pro zpracování informací a jejich zabezpečení včetně zajištění důvěrnosti, ověřitelnosti a integrity informace během jejího přenosu, zpracování a uchovávání v počítačových systémech. [24]

Popsané kryptografické algoritmy v tomto standardu jsou navrženy pro hardwarové i softwarové implementace a vyhovují moderním kryptografickým požadavkům a neaplikují omezení na úrovně důvěrnosti zabezpečené informace. Standard je určen pro vývoj, operaci a modernizaci informačních systémů s různým využitím. [24]

2.1.6 GOST R 34.11-2012

GOTST R 34.11-2012 je dalším standardem ruské federace a modernizace předchozího standardu GOST R 34.11-94. Na rozdíl od standardu GOST 34.12-2015 tento standard definuje algoritmus a proceduru hašovací funkce použitelnou pro libovolnou sekvenci binárních symbolů. Dále popisuje oblast působnosti v kryptografických metodách datového zpracování a zabezpečení, včetně procedur digitálního podpisu pro přenos dat a jejich archivaci v počítačových systémech. Proces výpočtu hodnoty hašovací funkce je ve standardu popsán ve třech krocích: [23]

1. Vygenerování klíče o délce 256 bitů.
2. Samotná šifrovací transformace 64 bitových slov za použití klíčů.
3. „Míchací“ transformace k dokončení šifrování. [23]

2.1.7 OpenPGP

Standard OpenPGP využívá kombinaci relativně silné asymetrické a symetrické kryptografie k zabezpečení služeb elektronické komunikace a uchovávání dat. Tyto služby poskytují uživateli: důvěrnost dat, management hesel, ověření pravosti zprávy a digitální podpisy. Hlavními funkcemi tohoto softwaru jsou služby datové integrity pro zprávy a datové soubory využitím: digitálních podpisů, šifrování, komprese a převodu radix-64. Doplňkem je zde poskytování heslového managementu a služeb pro certifikáty. Standard OpenPGP dále popisuje výběr symetrických, asymetrických, hašovacích i kompresních algoritmů včetně dalších metod a standardů (např. metoda derivace hesla S2K). [27] [28]

Tabulka 1 Kryptografické metody standardu OpenPGP [28].

Asymetrické:	RSA, DSA, Elgamal
Symetrické:	IDEA, TripleDES, Cast5, Blowfish, AES, Twofish
Hašovací funkce:	MD5, SHA-1, RIPE-MD/160, SHA (256,384,512/224)
Kompresní metody:	ZIP, ZLIB, BZip2

2.1.8 ISO/IEC 18033-3:2010

ISO (Mezinárodní Organizace pro Standardizace) a IEC (Mezinárodní Elektrotechnická komise) spolu formují specializovaný systém pro celosvětovou standardizaci. Tento standard byl zhotoven komisí JTC 1 pro informační technologie a komisí SC 27 IT bezpečnostní technologie. Standard popisuje šifry v následující tabulce: [61]

Tabulka 2 Šifrovací algoritmy standardu ISO/IEC 18033-3:2010 [61]

Délka bloku	Název šifry	Délka klíče
64 bitů	TDEA	128 nebo192 bitů.
	MISTY1	128 bitů.
	CAST-128	
	HIGHT	
128 bitů	AES	128, 192 nebo 256 bitů.
	Camellia	
	SEED	128 bitů.

2.1.9 Doporučení BSI 2018

Tento dokument od německého Úřadu pro Bezpečnost Informace popisuje zhodnocení bezpečnosti a dlouhodobou orientaci u vybraných kryptografických mechanismů. V dokumentu jsou aktuální zhodnocení a trendy pro následující kryptografické metody: Symetrické a asymetrické šifrování, hašovací funkce, autentizace dat, utajené sdílení, generátory náhodných hodnot, autentizace relací. [65]

V kapitole symetrických šifrovacích schémat doporučuje využívání či přechod na blokové šifry o délce bloku 128 bitů a to především na AES s délkou klíče 128, 192 a 256 bitů a tento standard srovnává s šiframi Serpent a Twofish. Pří porovnávání těchto tří šifrovacích algoritmů dochází k závěru, že u všech tří se nenachází žádný negativní nález v oblasti bezpečnosti, ale ze všech zmíněných algoritmů byl nejvíce prověřen a analyzován AES. Tuto skutečnost uplatňuje pro klasické krypto-analytické útoky i pro další bezpečnostní aspekty, do kterých se řadí i odolnost vůči útokům v rámci specifických implementací. [65]

3 MODERNÍ METODY ZÍSKÁNÍ ZAŠIFROVANÝCH DAT.

3.1.1 Metody vyhledávající klíč

K vyhledávání tajných klíčů slouží množství metod a algoritmů, jejichž výběr závisí na použité hašovací funkci a rychlosti kalkulace tohoto otisku hesla. Zásadními faktory ovlivňující efektivitu či aplikovatelnost jedné z těchto metod závisí na čase, výkonu a pravděpodobnosti úspěchu specifické metody útoku.[70]

3.1.1.1 Útok hrubou silou

Útok hrubou sílou spočívá ve hledání hodnoty hašovací funkce (otisku hesla), výpočtem jeho hodnoty z každé možné kombinace znaků, pro předem určenou délku textového řetězce. Kompletní útok tohoto typu zaručuje 100% úspěšnost, avšak čas vynaložený na jeho uskutečnění při jeho realizaci je závislý na délce hledaného hesla a po přesáhnutí délky, typicky 10 znaků, je kalkulace této hodnoty díky obrovské délce klíčového prostoru časově nemožné (viz tabulky klíčového prostoru šifer a příloha P1). [70] [71]

3.1.1.2 „Mask“ útok

Tento druh útoku je vylepšeným útokem hrubou silou. Jeho vylepšení spočívá ve zmenšení klíčového prostoru, kterým musí algoritmus projít a to tím, že upravuje vlastnosti hledaných kombinací, které jsou bližší heslu vytvořeným běžným uživatelem. [70] [71]

3.1.1.3 Slovníkový útok

Při tomto útoku je využíván tzv. slovník, což je soubor obsahující tisíce slov a hesel, kde je každá hodnota kandidátem na hledanou hodnotu. Rozdíl mezi tímto útokem a hrubou silou je, že tento druh útoku hledá shodu v konečné množině kombinací znaků, předem zapsaných v souboru, zatímco útok hrubou silou prochází každou kombinací v daném klíčovém prostoru. [70] [71]

3.1.1.4 „Rainbow“ tabulkový útok

Tento útok využívá před-vypočítanou tabulku pro získání správného otisku hesla. Každý set těchto tabulek je vytvořen pro specifickou velikost hesla obsahující předem určenou posloupnost znaků. Cílem této metody je zmenšení doby, která by byla vynaložena k tvorbě hašovací hodnoty, při útoku hrubou silou. Při takovémto druhu útoku jsou velké požadavky na paměť v závislosti na délce hledaného hesla. [70] [71]

3.1.1.5 Hybridní slovníkový útok

Tato metoda je kombinací útoku hrubou silou a slovníkovým útokem. Algoritmus používá jednotlivé slova slovníku a kombinuje je s řetězcem znaků. Tato varianta útoku avšak exponenciálně zvyšuje nároky na výkon a čas, který se odvíjí od množství znaků, které mají být připojeny k jednotlivým zápisům souboru slovníku. [70]

3.1.2 Metoda získání dat fyzickým přístupem k paměti.

Metoda spočívá v získání bitové kopie dat paměti a její následné analýze a to za pomocí speciálního forenzního hardwaru a softwaru. Mezi hardwarové nástroje pro získání kopie paměti slouží například metoda využívající sběrnici FireWire, která umožňuje přímý přístup k fyzické paměti avšak tato sběrnice je zastaralá a na moderních počítačích ji téměř nenajdeme. Dalším nástrojem je software KntDD, jenž se využívá pro zálohování a analýzu paměti systému Windows. Tento software je možné pustit z externího disku a zálohu je možné provést i přes síť. Také bezplatný software Belkasoft Live Ram Capturer lze spustit z externího disku a poskytuje nástroj pro analýzu paměti. [70]

3.1.3 Hashcat

Hashcat je bezplatný open-source software pro obnovu a získání hesel pomocí pokročilých kryptografických metod. Nejnovější verze 4.1 podporuje množství metod obnovy hesla na grafických kartách, procesorech, FGPA jádrech a dalším hardwaru podporujícím OpenCL. Dále podporuje využití více zařízení najednou, benchmarking (testování výkonu hašovacích funkcí na hardwaru) a přes 200 různých druhů hašovacích funkcí. Jádrem aplikace jsou 4 hlavní módy útoků: Slovníkový, kombinační, hrubé síly a hybridní. [64]

3.1.4 Passware Kit Forensic

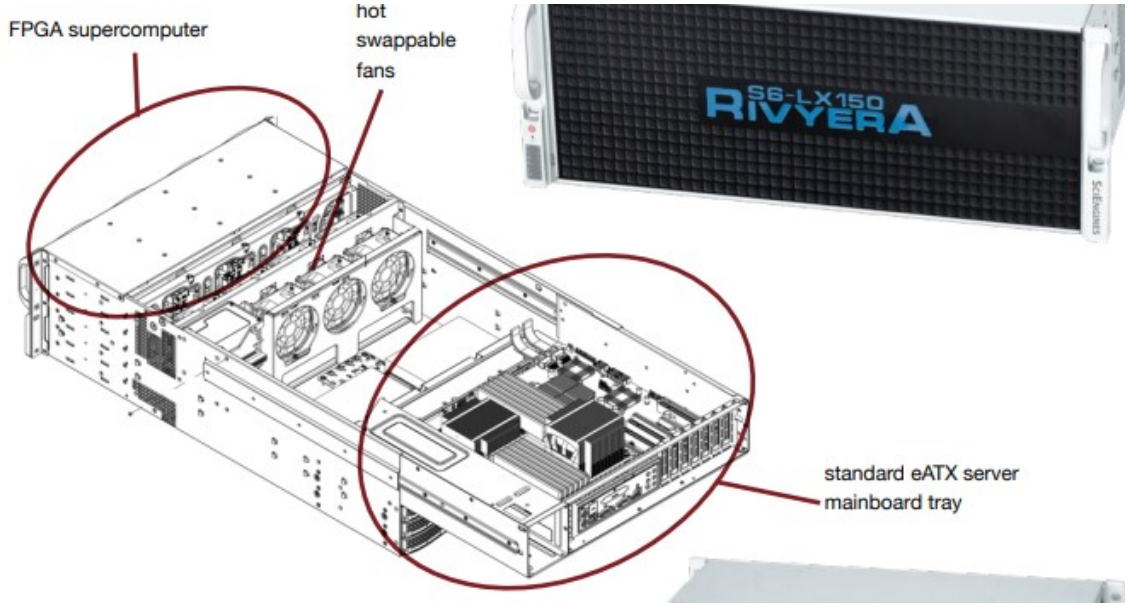
Tento software je balíčkem funkcí pro rozluštění hesla, analýzu šifrovaných dat, kontejnerů a disků. Poskytuje podporu pro více jak 280 druhů typů souboru, v tomto počtu jsou zahrnuty i jednotlivé verze těchto typů.[70]

Mezi hlavní funkce této aplikace patří:

- Detekování zašifrovaných souborů a kontejnerů.
- Extrahování šifrovacích klíčů a hesel z paměti.
- Využívá hardwarovou akceleraci.
- Získávání dat z cloudových aplikací. [70]

3.1.5 High-end krypto-analytický hardware: SciEngines Rivyera S6

SciEngines je malá společnost založena v roce 2007 a se sídlem v Kielu, která se zabývá velmi specializovaným odvětvím super-počítačů a vysoké výpočetní síly. Jejich nabídkou jsou pře-konfigurovatelné počítače s vysokým výkonem pro náročné výpočetní aplikace. Jejich řešením je kombinace FGPA technologie a masivně-paralelní architektury. [58]



Obrázek 12 - Xilinx Spartan-6 LX150

Serverová verze Xilinx Spartan-6 LX150 obsahuje ve své standartní konfiguraci 8 až 128 FGPA a eATX počítač, který slouží jako rozhraní se zbytkem sítě. Konfigurovatelné parametry nabízí až 65GB paměti DDR3 a 4TB paměti SDHC. Jednotka je napájena pomocí 1280W nebo 3000W zdroje. Cena jednotky záleží na počtu FGPA v konfiguraci (16 až 128), a pohybuje se přibližně 19 900-86 900 Euro (500 000-2 200 000 Kč) [58]

Tabulka 3 Výkon jednoho FGPA na Xilinx Spartan-6 LX150 (16 až 128 FGPA na 1 zařízení)[58]

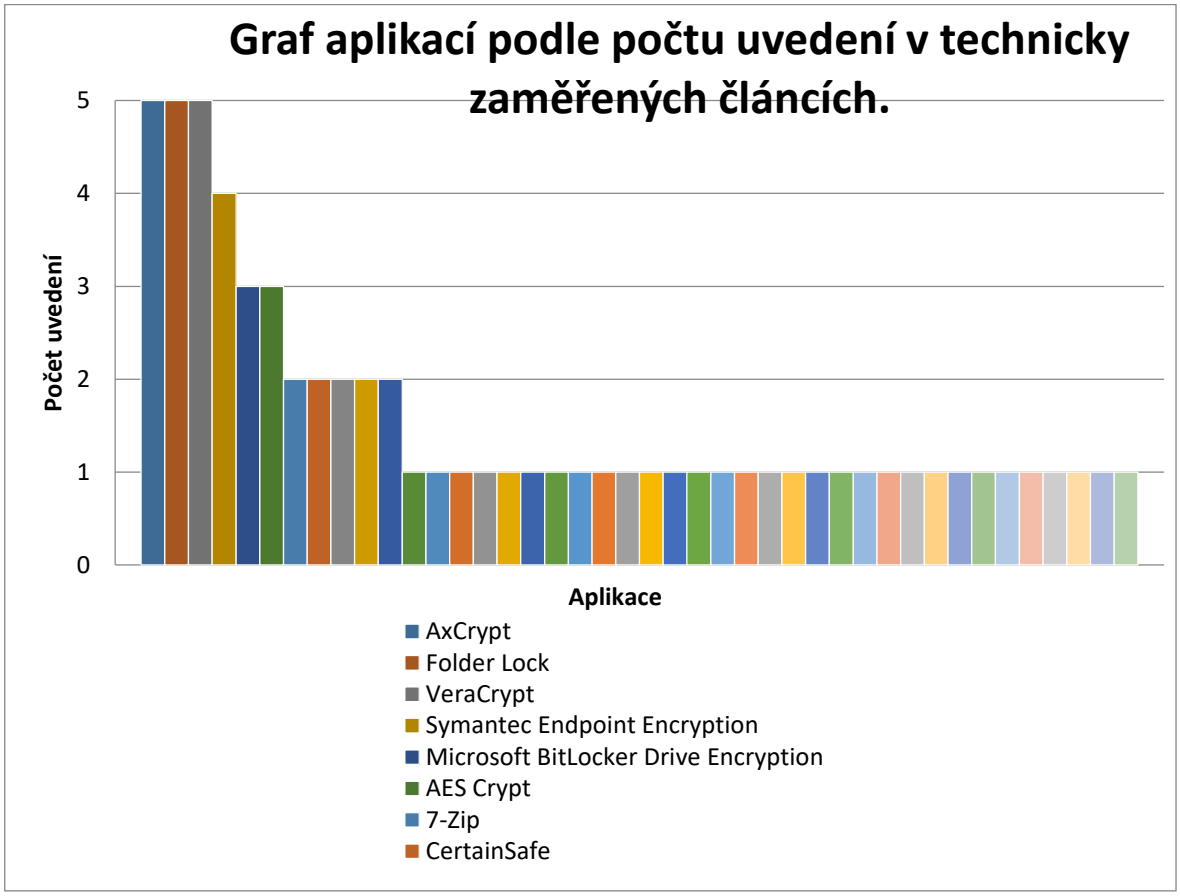
Algoritmus	Počet jader na jednom FGPA	Kmitočet v MHz	Rychlost v Mh/s	Max. délka hesla / klíče
SHA-1	3	125	375	55
SHA-256	1	175	175	55
SHA-512	1	100	50	55

Whirlpool	2	175	175	31
Ripemd-160	1	175	175	55
AES-256	4	150	600	32
Serpent-256	3	120	360	32

II. PRAKTICKÁ ČÁST

4 VÝBĚR KRYPTOGRAFICKÝCH METOD A APLIKACÍ PRO JEJICH PRAKTICKÉ OTESTOVÁNÍ.

Pro výběr kryptografických metod posloužily články technicky zaměřených webových stránek s výběry produktů pro šifrování dat 6 na sobě nezávislých stránek: PCMAG[33], Techradar.Pro [34], SYSTWEAK [35], Windows Central [36], TechViral [37] a eSecurity Planet [38].



Graf 1 Aplikace podle počtu zmínek.

Z výsledků shromážděných dat a grafu 1 (nahoře), kde se nachází celkem 42 různých produktů, můžeme vidět na prvních 3 příčkách: AxCrypt, Folder Lock, VeraCrypt, Synmatec Endpoint Encryption, Microsoft Bitlocker Drive Encryption a AES Crypt.

5 TESTOVACÍ HARDWARE

5.1.1 Intel Xeon E5-2630 v3 (Haswell)

Tento 32 jádrový procesor ve verzi se 128GB pamětí DDR4 od společnosti Intel je především určen pro servery a tomu napovídají i jeho technické detaily. Procesor byl vydán ve třetím čtvrtletí roku 2014, ale svým výkonem je stále aktuální. Kmitočet procesoru je až 3,2GHz a má k dispozici 20MB cache paměť. Maximální spotřeba procesoru je v zátěži 85W. Důležité je zde, že má k dispozici instrukční set AES-NI, což má zásadní dopad na rychlost výpočtových operací nad algoritmem AES. Cena v základním provedení je přibližně 14 205 Kč. [74]

5.1.2 Tesla K80

Tesla K80 je výkonná grafická karta využívána pro analýzu a kalkulaci vědeckých aplikací. Vnitřní struktura této grafické karty se ve skutečnosti skládá z dvojice grafický karet GK210 na mikro-architektuře nVidia Kepler 28nm a dohromady obsahuje 4 992 CUDA procesorů s taktem 562 až 875 MHz. V porovnání s dnešní vyšší třídou výkonných grafických karet se i tak na téměř 4 roky staré Tesle nachází přibližně dvakrát tolik procesorů. Dále je na grafické kartě celkem 24 GB paměti GDDR5. Maximální teoretická propustnost grafické karty se uvádí 8.7 teraflopů. Aktuální cena 129 556 Kč s DPH (web Heuréka). [42] [66]

5.1.3 Strix Asus RX460 4GB

Grafická karta RX460 patří do skupiny střední třídy, předposlední (4.) generace grafických karet založené na GCN architektuře od AMD. Kódový název GPU pro serii RX 460 je Polaris 11. Propustnost grafické karty je 2,2 TFLOP s příkonem pod 75W. Karta je vybavena 14 CU jednotkami, 896 stream procesory a 4GB pamětí typu GDDR5. Technologie výroby je 14nm. Aktuální min. cena 5847 kč s DPH (web Heureka). [55] [66]

5.1.4 Phenoix Gainward GTX 1070 (GP104-A)

Tato téměř 2 roky stará grafická karta pochází z nejnovější generace grafických karet nVidia na architektuře Pascal, která disponuje 16nm výrobní technologii a díky tomuto bylo možné zvýšit počet tranzistorů na 7,2 miliard s 1920 CUDA jádry, se základním taktem 1506 MHz. Stejně tak, jak je tomu u předchůdce této grafické karty, je grafické jádro rozděleno do 4 grafických clusterů (GPC) avšak čerpá z více zdrojů, což má za výsledek lepší

výkon oproti předchůdci (GTX 970). Grafická karta má 8GB paměti GDDR5. Aktuální min. cena 13 165 kč s DPH (web Heureka). [66][67]

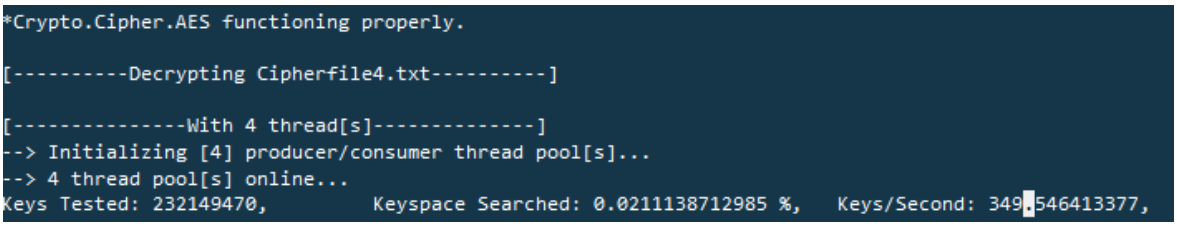
5.1.5 GT 540M

Grafická karta společnosti nVidia GeForce GT 540M patří do skupiny střední třídy mobilních karet pro laptopy. Jádro GPU s frekvencí 672Mhz je založeno na architektuře GF108 (96 CUDA jader) s výrobní technologií Fermi 40nm. Tato verze GF108 má 1GB paměti DDR3. Jako mobilní karta s menší spotřebou energie je avšak optimalizována pro skupinu spotřebovatelů s vyššími výkonnostními nároky (multimédia a hry). Cena 1836 Kč bez poštovného (web Amazon). [68]

6 POPIS KRYPTOGRAFICKÝCH IMPLEMENTACÍ A JEJICH PRAKTICKÉ OTESTOVÁNÍ

6.1.1 AES-128 výkonový test (CPU)

AES je jednoznačně nejpoužívanější symetrický algoritmus pro data-šifrující aplikace, který je v řadě standardů a kryptografických metod. To je i podpořeno faktem, že se nachází v každé ze zde popsaných aplikací (AxCrypt, Folder Lock, VeraCrypt, symantec, Symantec E. D., M. Bitlocker a AES Crypt), jenž byly vybrány na základě dat z technicky zaměřených webových stránek a magazínů.



Obrázek 13 Snímek obrazovky s aplikací AES cracker-128 na fakultním serverovém procesoru viz použitý hardware (kapitola 5).

Na obrázku 12 je snímek obrazovky z výkonového testu algoritmu AES se 128 bitovým klíčem na fakultním serverovém procesoru (v předchozí kapitole). V obrázku vpravo dole si můžeme všimnout rychlosti hledání klíčů (~349~~,55~~ klíčů za sekundu). Touto rychlostí bychom prohledali celý 128 bitový klíčový prostor tohoto algoritmu za přibližně $3{,}16993\times10^{28}$ let.

6.1.2 AxCrypt

AxCrypt je jednou z předních open-source šifrovacích aplikací. Je to nástroj pro kompresi, šifrování, ukládání, odesílání a práci s individuálními soubory a tedy nepodporuje celodiskové šifrování. K šifrování dat využívá šifru AES s délkou klíče 128 nebo 256 bitů, v CTR módu a asymetrické šifry RSA-4096 pro zašifrování hlavního klíče k zabezpečení sdílení souborů a ke změně uživatelského hesla. Souborový formát je navržen s rezervou tak, aby umožnil funkci blokovým šifrám o délce bloku do 256 bitů a proudovým i blokovým šifrám s klíčovým prostorem do 512 bitů. [29] [30]

6.1.2.1 Šifrovací schéma

Obrázek 14 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace AxCrypt V2.

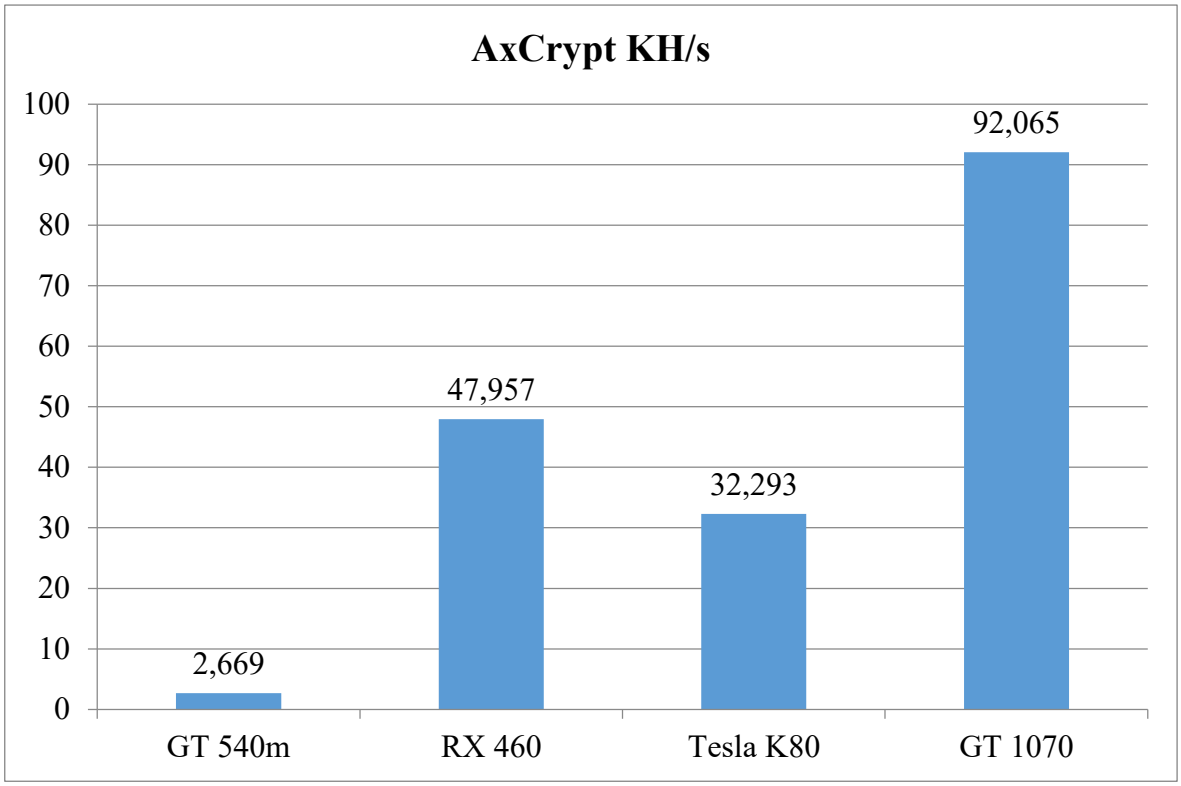
1. Klíč balící klíč (KEK): Klíč je derivován pomocí algoritmu PBKDF2-HMAC-SHA512. Výsledkem derivace je 64 bytů a následná redukce na požadovanou délku klíče. Tyto data jsou použity jako klíč pro klíč-balící algoritmus k zabezpečení hlavního šifrovacího klíče. [30]
2. Hlavní klíč (MEK): Náhodně generovaný klíč využit k zašifrování samotných dat a ke kalkulaci ověřovací HMAC hodnoty. MEK je zabezpečen balícím algoritmem standardu NIST, který používá KEK jako klíč. Pro generování klíče využívá pseudo-náhodný generátor čísel poskytnutý operačním systémem. [30]

Operační mód symetrické šifry je CTR, což má za výsledek to, že se zde bloková šifra chová jako proudová. Důvodem použití tohoto operačního módu je, že není třeba dalšího odsazování mezi bloky dat.[30]

Tabulka 4 Přehled šifrovacích algoritmů aplikace Axcrypt.

Algoritmus.	Délka klíčového prostoru.
AES 128/256 (CTR)	2^{128} ($3{,}4\times10^{38}$), 2^{256} ($1{,}15\times10^{77}$)
SHA-512	2^{512}($1{,}34\times10^{154}$)
RSA-4096	2^{4096} ($1{,}04\times10^{1233}$)

6.1.2.2 Výkonový test (Hashcat benchmark)



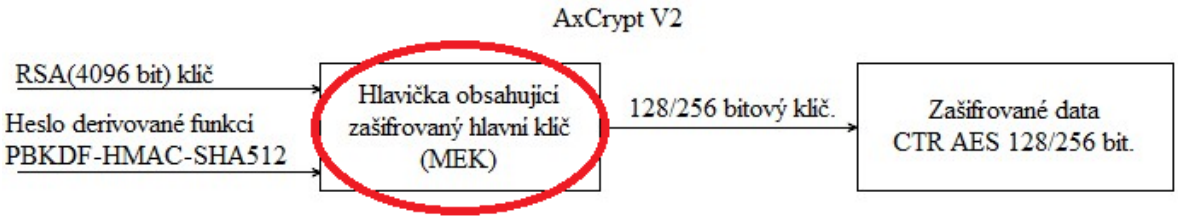
Graf 2 Porovnání výsledků výkonových testů heslo-derivující funkce aplikace AxCrypt podle grafické karty.

Tabulka 5 Počet dní potřebných k vykonání útoku hrubou silou (GT 1070) v závislosti na délce hesla, znakové sadě a použité derivační funkci (v grafu 2).

Znaková sada / Délka hesla	**26**	**36**	**52**	**62**	**96**
4	0	0	0	0	0
5	0	0	0	0	1
6	0	0	2	7	98
7	1	9	129	442	9446
8	26	354	6720	27448	906904
9	682	12767	349479	1701832	87062833

6.1.2.3 Místo testování

Pokud porovnáme rychlosti hledání klíčů AES (obrázek 12) a hesel funkce AxCrypt (graf 2) a za předpokladu, že hlavní klíč je vygenerován náhodně a nejde zpětně odvodit (což zaručuje generátor náhodných hodnot operačního systému), bude nejvhodnějším místem útoku na balící algoritmus hlavního klíče, kde není náhodnost vždy zaručena. Bezpečnost bude potom záviset na síle a délce hesla, které zadal samotný uživatel. [30]

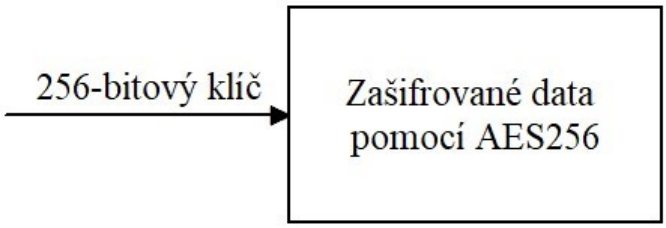


Obrázek 15 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace AxCrypt V2 s zvýrazněným místem testování.

Testování pomocí aplikace Hashcat však nebylo úspěšné z důvodu nekompatibilní verze souborového kontejneru AxCrypt V2 s verzí aplikace Hashcat 4.1.0.

6.1.3 Folder Lock

Tento software pro zabezpečení dat nabízí uzamčení, skrytí a zabezpečení souborů heslem s jednoduchou interakcí pomocí drag-drop. Hlavní funkce této aplikace jsou: Uzamčení souborů, šifrování soborů, zabezpečení zálohy, zabezpečení USB/CD, uložení finančních údajů, skartování souborů. K šifrování používá symetrickou šifru AES-256.[31]



Obrázek 16 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace Folder Lock.

Přes veškerou snahu navázat kontakt s autory této aplikace nebylo možné získat další informace o implementaci použitých kryptografických metod a standardů.

6.1.4 VeraCrypt

VeraCrypt je bezplatný, open-source software pro šifrování disku na operačních systémech Windows, Mac OSX a Linux. Je založen na známém šifrovacím softwaru TrueCrypt7.1a. Mezi hlavní funkce VeraCryptu patří: Vytvoření virtuálního šifrovaného disku ve formě souboru, který je následně používán jako fyzický disk; Šifrování celé jednotky nebo uložiště (HDD nebo USB); Poskytuje šifrování automatické, v běhu a transparentní; Hardwarově akcelerované šifrování; Stenografii a skrytý operační systém. Software aktuálně nabízí šifrování pomocí blokových symetrických 256 bitových algoritmů AES, Camellia, Kuznyechik, Serpent, Twofish v módu XTS a případně kaskádu kombinací z předešle zmíněných algoritmů. [32]

6.1.4.1 Šifrovací schéma

Obrázek 17 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace VeraCrypt

1. Klíč hlavičky (Header Key): Zabezpečuje data hlavičky, které obsahují hlavní klíč.
2. Hlavní klíč (MEK): Slouží pro přístup k samotným datům a je generován pomocí náhodných vstupních dat (pohybu myši uživatele) při tvorbě oddílu.

Derivace klíče (podle standardu PBKDF2) probíhá z následujících parametrů a vstupů:

- Heslo - jenž je vstupem uživatele.
- PIM hodnota – nastavitelná hodnota určující počet iterací hašovací funkce.
- Sůl - náhodně generované data.

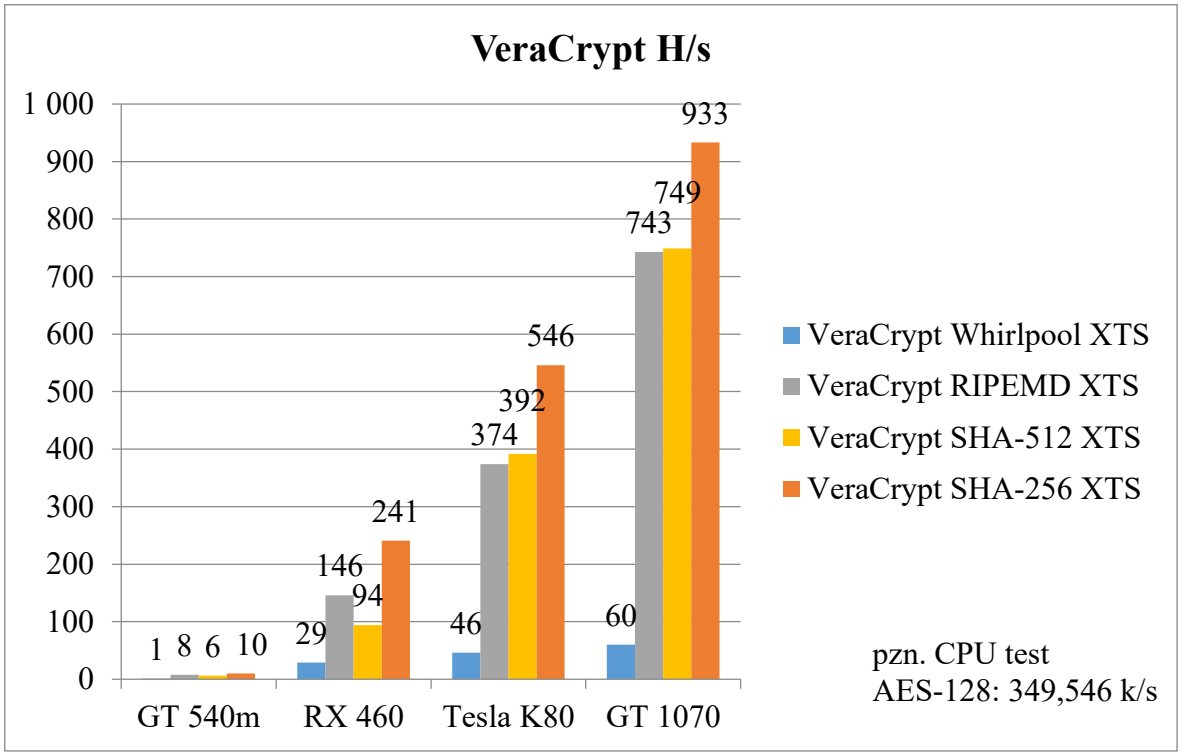
K derivaci klíče hlavičky z uživatelského hesla využívá následující hašovací funkce v kombinaci s ověřením HMAC: SHA-512, Whirpool, SHA-256 a RIPEMD-160. Klíče

jsou vždy stejné velikosti, nezávisle na použité hašovací funkci. Počet iterací aplikovaných pro systémové šifrování, které využívá hašovací funkci SHA-512 nebo Whirpool je počet iteraci roven PIM×2048. Pro systémové, nesystémové šifrování a souborové kontejnery je pro hašovací funkce RIPEMD-160 a SHA256 počet iterací 15000 + (PIM×1000) [32]

Tabulka 6 Přehled šifrovacích algoritmů aplikace VeraCrypt.

Algoritmus.	Délka klíčového prostoru.
AES, Serpent, Twofish, Camellia, Kuznyechik 256 (XTS)	2^{256} ($1{,}15\times10^{77}$)
RIPEMD-160	2^{160}($1{,}46\times10^{48}$)
SHA-256	2^{256} ($1{,}15\times10^{77}$)
Whirlpool	2^{512} ($1{,}34\times10^{154}$)
SHA-512	2^{512}($1{,}34\times10^{154}$)

6.1.4.2 Výkonový test (Hashcat benchmark)



Graf 3 Porovnání výsledků výkonových testů heslo-derivujících funkcí aplikace VeraCrypt podle grafické karty.

Tabulka 7 Počet dní potřebných k vykonání útoku hrubou silou (GT 1070) v závislosti na délce hesla, znakové sadě a použité derivační funkci (VeraCrypt SHA-256 XTS).

Znaková sada / Délka hesla	**26**	**36**	**52**	**62**	**96**
3	0	0	0	0	0
4	0	0	0	0	1
5	0	0	4	11	101
6	3	27	245	704	9710
7	99	972	12753	43686	932187
8	2590	34996	663179	2708557	89489993

6.1.4.3 Místo testování

S náhodně generovanými hlavními klíči (pomocí pohybu myši) při tvorbě oddílu se šance nalezení hlavních klíčů (viz operační mód XTS a graf 3) velmi zmenšuje, proto i zde je nejatraktivnější pro potencionálního útočníka zkusit napadnout slabé uživatelské heslo.

Obrázek 18 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace AxCrypt V2 s zvýrazněným místem testování.

6.1.4.4 Testování oddílu VeraCrypt

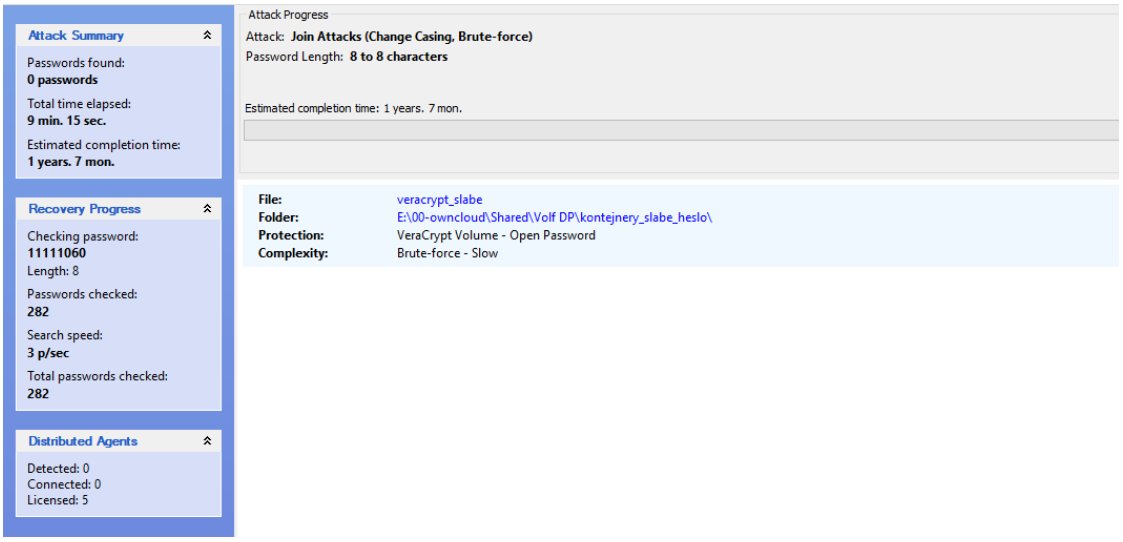
Pro testování byl vytvořen virtuální oddíl VeraCrypt s velmi slabým heslem: apple123. Oddíl byl vytvořen se základními parametry: Výchozí počet iterací derivační funkce PBKDF-HMAC-SHA256, symetrická šifra AES, souborový systém FAT s velikostí oddílu 1MB. Tento kontejner byl otestován mask útokem (viz druhy útoků) s parametrem známé velikosti hesla a přibližné podoby (5 znaků malých písmen + 3 číslice), což snížilo čas po-

třebný na kompletní vykonání testu na pouhých 323 dní (viz tabulka 6). Vzhledem k podobě hesla bylo nalezeno v první 0,01% klíčového prostoru tedy asi po 1 hodině testování.

Pod textem je výstup dokončeného testování z okna aplikace Hashcat 4.1.0

```
Session..........: hashcat 4.1.0
Status...........: Cracked
Hash.Type........: VeraCrypt PBKDF2-HMAC-SHA256 + XTS 512 bit
Hash.Target......: VOLF-DP/veracrypt_slabe
Time.Started.....: Wed May  2 07:53:32 2018 (1 hour, 10 mins)
Time.Estimated...: Wed May  2 09:03:47 2018 (0 secs)
Guess.Mask.......: ?l?l?l?l?l?d?d?d [8]
Guess.Queue......: 1/1 (100.00%)
Speed.Dev.#1.....:      221 H/s (6.64ms) @ Accel:16 Loops:4 Thr:896 Vec:1
Speed.Dev.#2.....:      204 H/s (7.22ms) @ Accel:16 Loops:4 Thr:896 Vec:1
Speed.Dev.#*.....:      425 H/s
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 1677312/11881376000 (0.01%)
```

Tento zašifrovaný oddíl byl nadále testován v aplikaci Passware a jak je vidět z obrázku 19 čas pro vykonání tohoto testu je pro tento účel velmi nepraktický.



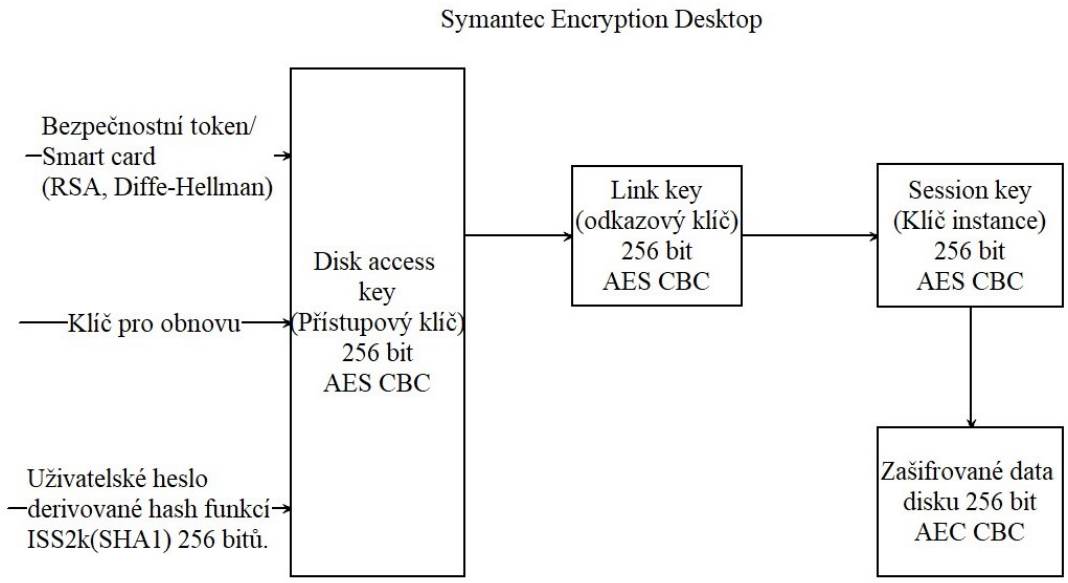
Obrázek 19 Testování kontejneru VeraCrypt pomocí softwaru Passware forensic Kit.

6.1.5 Symantec Endpoint Encryption – Encryption Desktop

Tento balíček aplikací pro šifrování umožňuje koncové, emailové a souborové šifrování. Software je určen především pro celkové podnikové řešení, ale lze si pořídit i jen jeho jednotlivé části. Symantec nabízí samostatnou aplikaci pro šifrování disku a souborů pod názvem Encryption Desktop, která je založena na předchůdci PGP Whole Disk encryption. Aplikace využívá PGP generované asymetrické klíče (RSA) a Diffe-Hellman s volitelnou velikostí od 1024 do 4096 bitů k zašifrování symetrického 256 bitového klíče AES v módu

CBC. Spolu s šifrováním jednotlivých diskových oddílů aplikace nabízí i šifrování souborových adresářů a vytváření šifrovaných virtuálních disků. [38] [41] [52]

6.1.5.1 Šifrovací schéma



Obrázek 20 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace Symantec Encryption Desktop.

Během procesu šifrování dat využívá aplikace 3 druhy klíče.

1. Přístupový klíč disku (Disk access key) je zašifrovaný pomocí veřejného klíče bezpečnostního tokenu nebo uživatelského hesla.
2. Odkazový klíč (Link key) je zašifrovaný pomocí přístupového klíče a spojuje jednotlivé disky dohromady.
3. Instanční klíč (session key), je zašifrován pomocí odkazového klíče a slouží jako přístup k samotným blokům dat. [57]

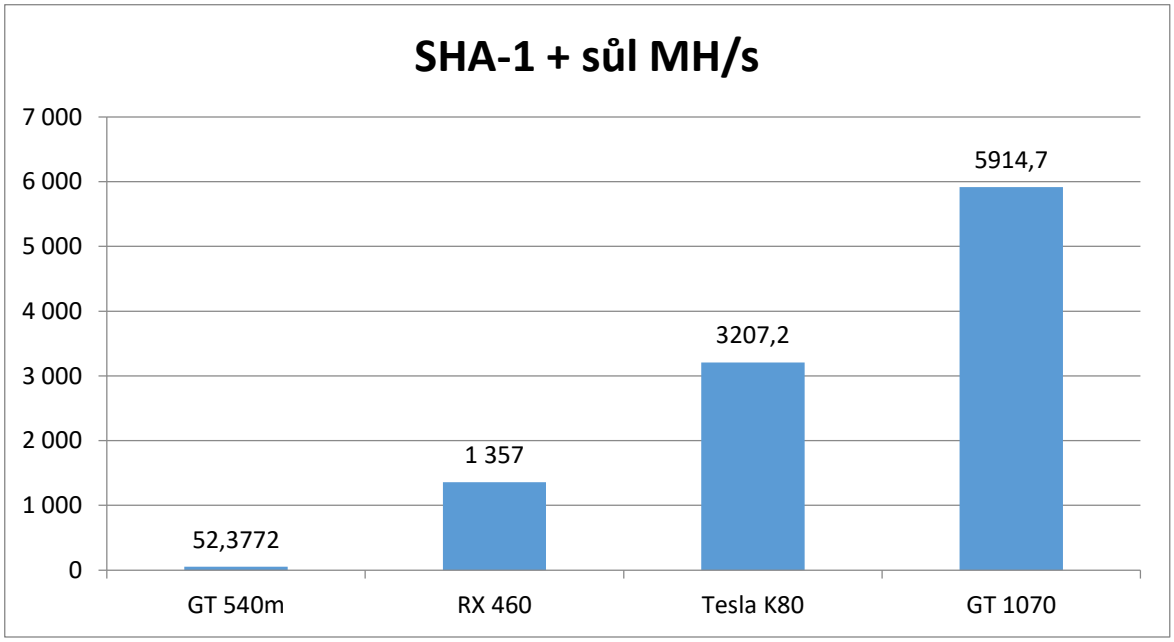
Uživatelské heslo, v procesu diskového šifrování, je derivováno pomocí metody nazývané ISS2K (Klíč z iterovaného, soleného řetězce), která je popsaná ve standardu OpenPGP RFC 4880 [28]. [57]

Tabulka 8 Přehled šifrovacích algoritmů aplikace Symantec Encryption Desktop.

Algoritmus.	Délka klíčového prostoru.
AES256 (CBC)	2^{256} ($1{,}15\times10^{77}$)

SHA-1	$2^{64}(1{,}84\times10^{19})$
RSA, Diffe-Hellman	$2^{1024\text{-}4096}$ $(1{,}79\times10^{308}\text{-}1{,}04\times10^{1233})$

6.1.5.2 Výkonový test (Hashcat benchmark)



Graf 4 Porovnání výsledků výkonových testů heslo-derivující funkce SHA-1 se solí podle grafické karty.

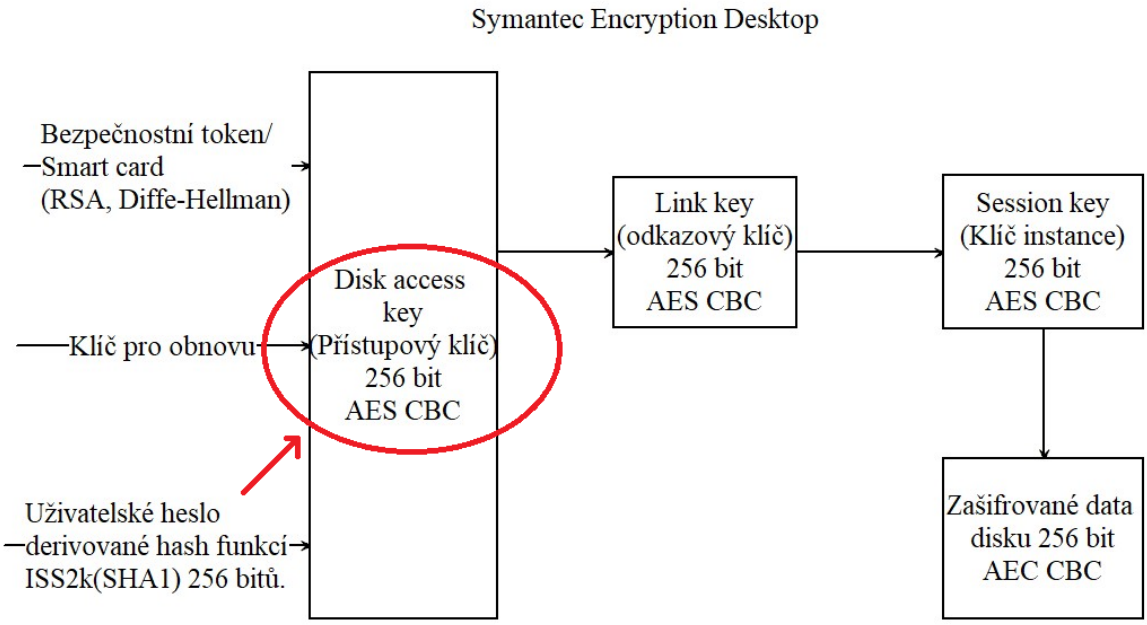
Tabulka 9 Počet dní potřebných k vykonání útoku hrubou silou (GT 1070) v závislosti na délce hesla, znakové sadě a použité derivační funkci (SHA-1 se solí).

Znaková sada / Délka hesla	**26**	**36**	**52**	**62**	**96**
7	0	0	0	0	0
8	0	0	0	0	14
9	0	0	5	26	1355
10	0	7	282	1642	130096
11	7	257	14709	101826	12489271
12	186	9272	764880	6313261	$1{,}2\times10^{9}$

13	4855	333799	39773792	$3{,}91\times10^{8}$	$1{,}15\times10^{11}$

6.1.5.3 Místo testování

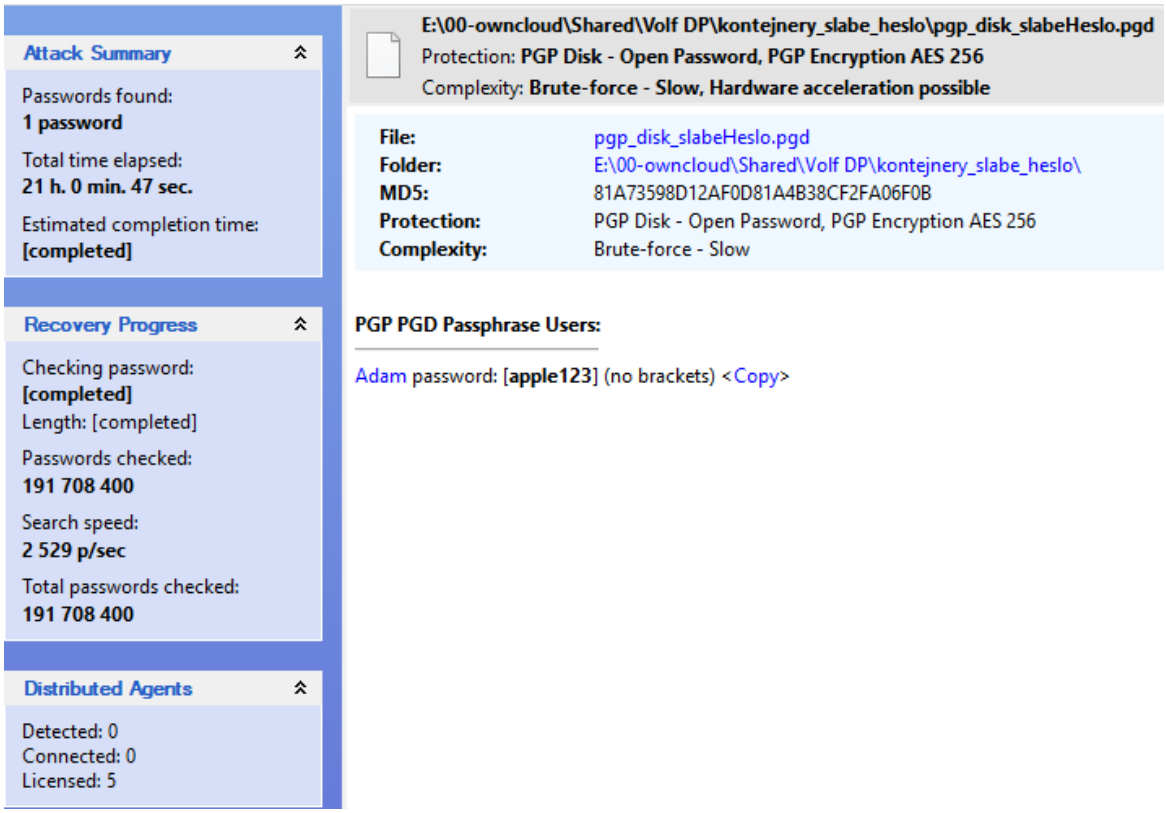
Symantec Encryption Desktop je primárně určen pro podnikové zabezpečení a disponuje více nástroji pro ochranu dat uživatele. Pro autentizaci uživatele lze využít bezpečnostní token, smart card i TPM čip, což při určitých scénářích velmi zužuje možnosti prolomení bezpečnosti šifrovaného oddílu. V grafu 4 je zobrazena rychlost generace hašovacích hodnot funkcí SHA-1 se solí, která je využívána při derivaci klíče z hesla pomocí metody ISS2K. [57]



Obrázek 21 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace AxCrypt V2 se zvýrazněným místem testování.

6.1.5.4 Testování kontejneru PGP

Pro praktické otestování byl vytvořen PGP kontejner s výchozím nastavením, souborovým systémem FAT, symetrickou šifrou AES256, velikostí 1MB a velmi slabým heslem: apple123. Tento kontejner byl otestován pomocí forenzního softwaru Passware Forensic Kit na fakultním počítači s procesorem (Intel Core i7-5820K). Heslo bylo nalezeno po 21 hodinách algoritmu útoku hrubou silou s rychlostí 2 529 klíčů za sekundu a prohledaným klíčovým prostorem o velikosti 191 708 400 klíčů (viz obrázek 20).

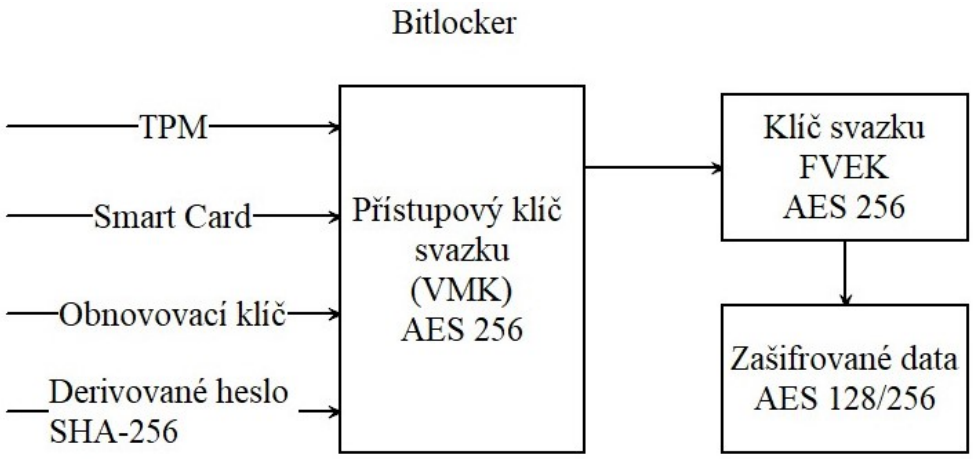


Obrázek 22 Snímek obrazovky z programu Passware Forensic Kit.

6.1.6 Microsoft Bitlocker Drive Encryption

Bitlocker je dnes základním vybavením každého operačního systému Windows a slouží k šifrování jednoho nebo více diskových jednotek. Tento základní nástroj systému Windows používá šifrovací algoritmus AES 128/256 v módu CCM a CBC nebo XTS a navíc také asymetrickou 2048 bitovou šifru RSA (ve formě TPM). [39] [40]

6.1.6.1 Šifrovací schéma



Obrázek 23 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace Microsoft Bitlocker.

Hlavní cílem Bitlockeru je zabezpečení dat uživatele, na svazku, na kterém se nachází operační systém. Ke splnění tohoto cíle jsou diskové sektory vždy zašifrovány pomocí dvou klíčů, které jsou zašifrovány algoritmem AES v módu CCM. [39] [40]

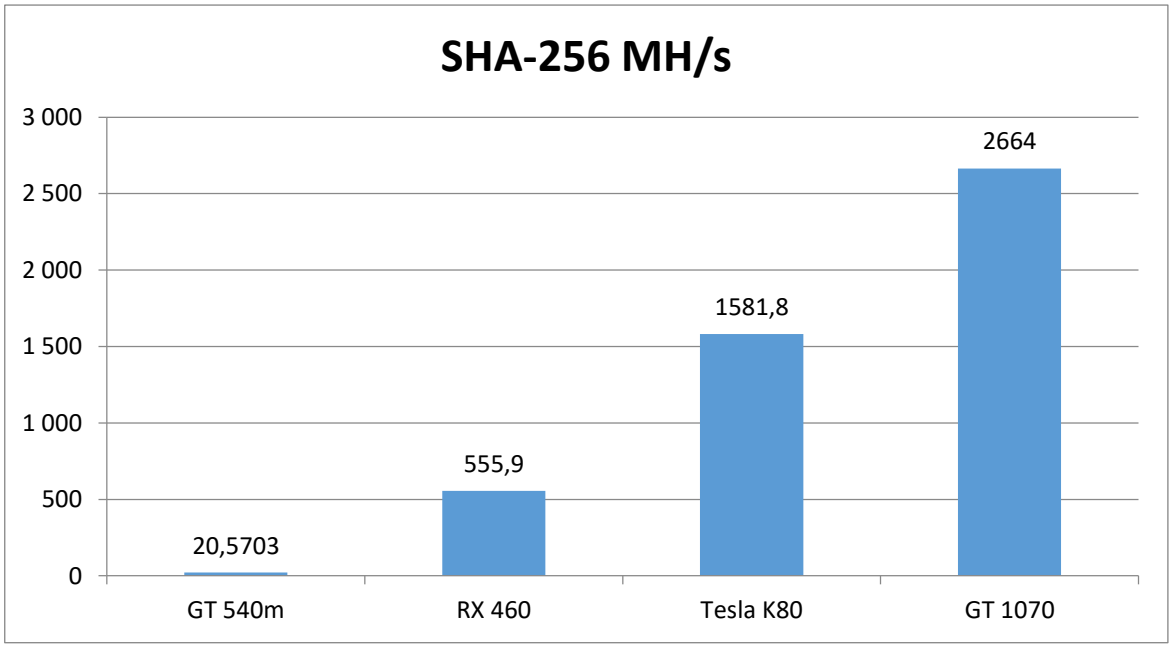
1. FVEK (Klíč svazku), který je vždy zašifrován klíčem VMK.
2. VMK (Přístupový klíč svazku), který zašifrován podle zvolených metod. [39] [40]

Dále je klíč VMK zašifrován heslem uživatele a zvolenou metodou autentizace TPM, Smart Card. Také lze vygenerovat obnovovací klíč o velikost 48 znaků, kterým lze odemknout VMK.[39][40]

Tabulka 10 Přehled šifrovacích algoritmů aplikace Microsoft Bitlocker.

Algoritmus.	Délka klíčového prostoru.
AES 128/256 (CBC, CCM, XTS)	2^{128} ($3{,}4\times10^{38}$), 2^{256} ($1{,}15\times10^{77}$)
SHA-256	2^{256} ($1{,}15\times10^{77}$)
RSA-2048	2^{2048}($3{,}23\times10^{616}$)

6.1.6.2 Výkonový test (Hashcat benchmark)



Graf 5 Porovnání výsledků výkonových testů hašovací funkce SHA-256 podle grafické karty.

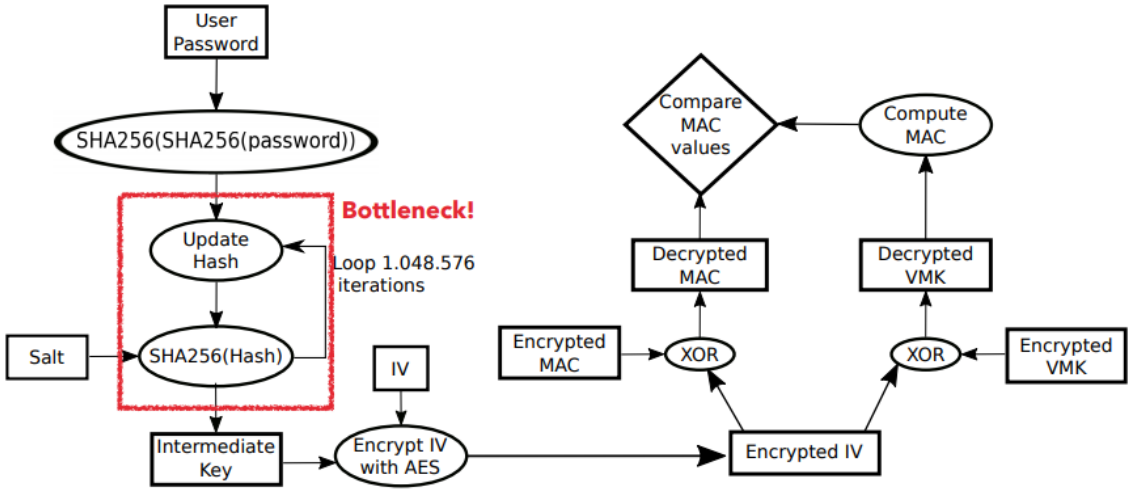
Tabulka 11 Počet dní potřebných k vykonání útoku hrubou silou (Tesla K80) v závislosti na délce hesla, znakové sady a použité hašovací funkci (Graf 5).

Znaková sada / Délka hesla	**26**	**36**	**52**	**62**	**96**
4	0	0	0	0	0
5	0	0	3	7	66
6	2	17	161	463	6389
7	65	639	8391	28744	613350
8	1704	23026	436351	1782147	58881639

6.1.6.3 Deepsec analýza softwaru Bitlocker

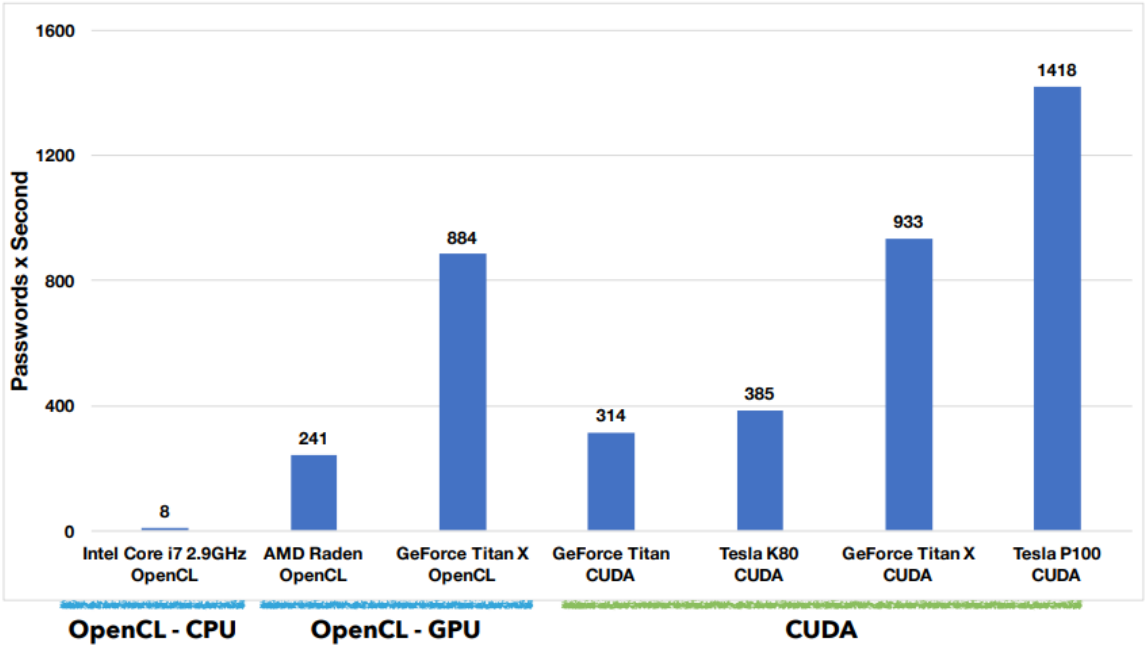
Tato analýza Bitlockeru se soustředí na autentizaci uživatelským heslem a využívá k tomu forenzní software Bitcracker, jehož algoritmus generace hesel je na obrázku číslo 15. Testování hesel pomocí slovníkového útoku zpomaluje 1 048 576 iterací hašovací funkce SHA-256, což má za výsledek pouhých 385 testovaných hesel za sekundu na grafické kartě

Tesla K80. Na obrázku 16 je algoritmus derivace a testování uživatelského hesla pomocí softwaru Bitcracker, kde je červeně označeno místo bottlenecku (místo, kde je náročná výpočetní operace), jenž je výsledkem 1 048 576 iterací hašovací funkce SHA-256. Při porovnání s vlastními výsledky výkonových testů Tesly K80, je tato rychlost po vydělení dříve uvedeným počtem iterací) přibližně 3,7 vyšší. Tento spíše zanedbatelný rychlostní rozdíl lze odůvodnit tím, že vlastní test proběhl pouze na hašovací funkci se solí oproti složitému algoritmu na obrázku 16.[40]



Obrázek 24 Algoritmus aplikace Bitcracker, jenž slouží k hledání uživatelského hesla k disku, který je zašifrován M. Bitlockerem. [40]

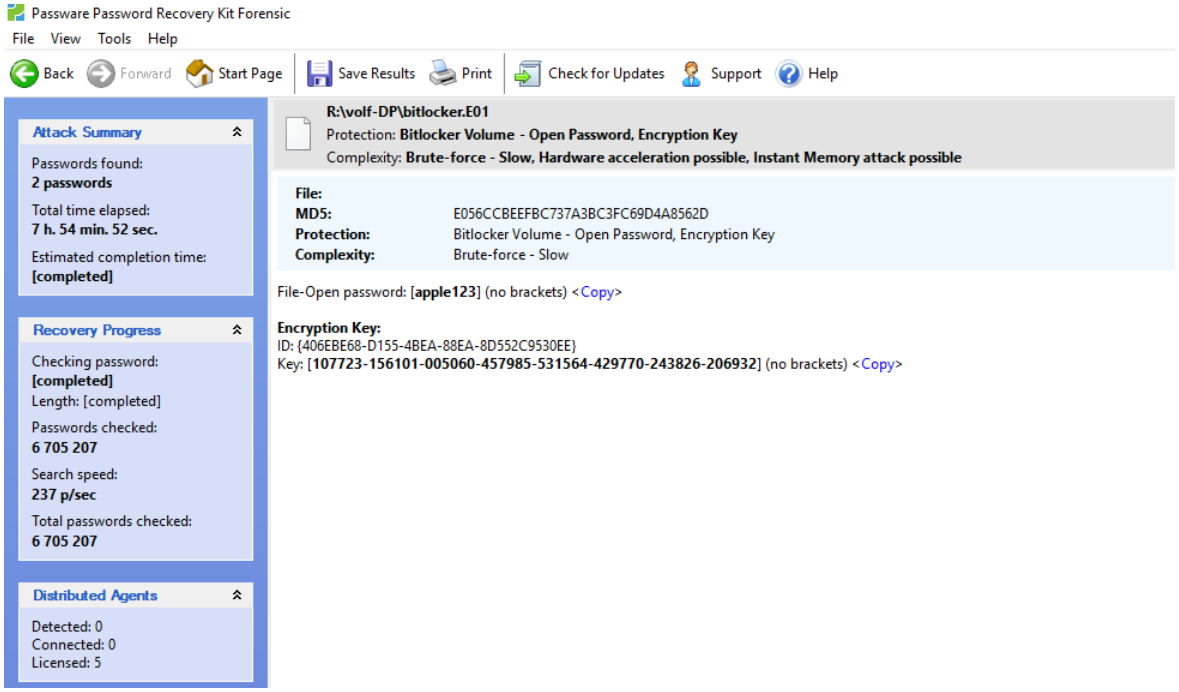
V obrázku 22 jsou zobrazeny výsledné rychlosti v softwaru Bitcracker, které jsou rozděleny podle typu hardwaru (GPU, CPU a CUDA jader) [40]



Obrázek 25 Porovnání rychlosti hledání hesel k algoritmu na obrázku 21 pomocí aplikace Bitcracker. (Podle druhu hardwaru)[40]

6.1.6.4 Testování softwaru Bitlocker

Podle obrázku 23 výše je složitost výpočetních operací pro nalezení hesel podobné složitosti, jako je tomu u softwaru VeraCrypt, avšak Bitlocker poskytuje navíc zabezpečení pomocí smart card nebo čipu TPM. Pro testování M.Bitlockeru byl vytvořen diskový obraz oddílů zabezpečeného tímto softwarem s identicky slabým heslem jako u předchozích testů (apple123). Pomocí forenzního softwaru Passware byl Bitlocker otestován mask útokem a po téměř 8 hodinách bylo nalezeno heslo (viz obrázek 25).

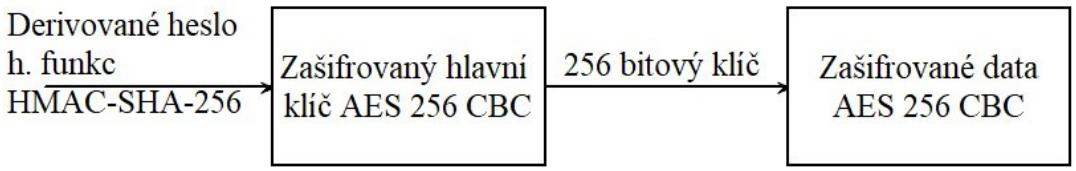


Obrázek 26 Snímek obrazovky z testování zašifrovaných dat pomocí M. Bitlocker.

6.1.7 AES Crypt

Tento šifrovací software je dalším bezplatným open-source projektem, který slouží k bezpečnému šifrování souborů. Je dostupný pro operační systémy Windows, Mac a Linux. Jak napovídá samotný název softwaru jeho šifrovacím algoritmem je AES. [56]

6.1.7.1 Šifrovací schéma



Obrázek 27 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace AES Crypt.

AES Crypt využívá k šifrování dva různé klíče:

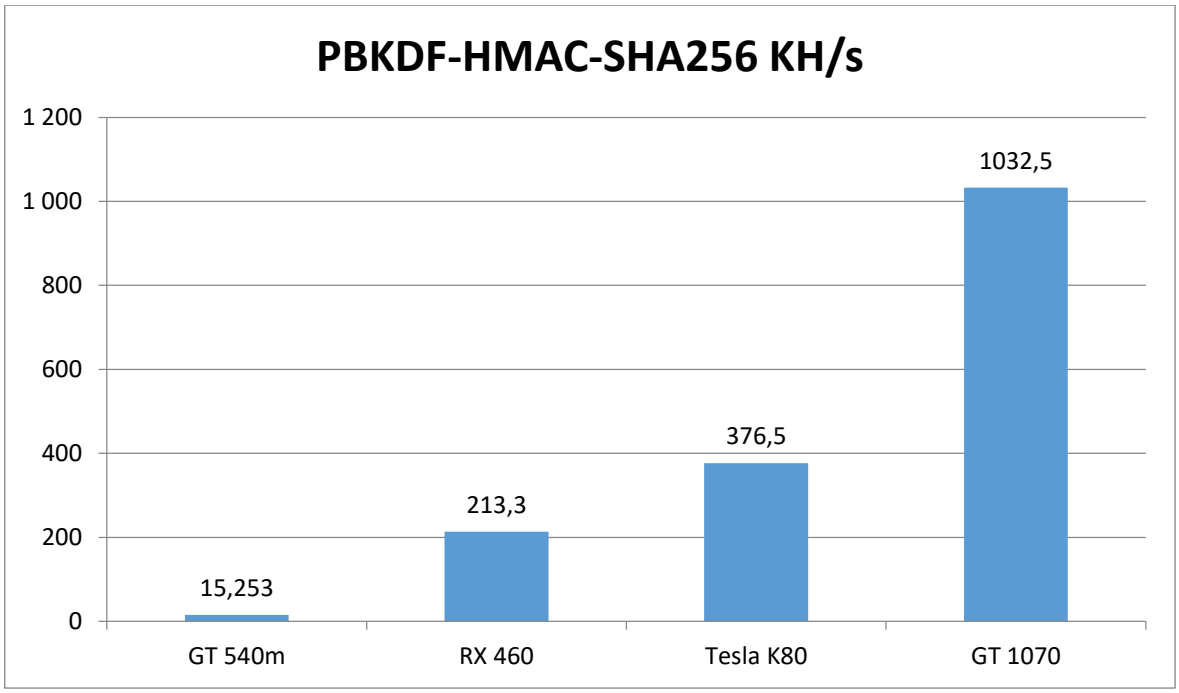
1. Klíč 1 (K-1). Klíč je derivován z uživatelského hesla 8192 cykly hašovací funkce SHA-256. Dále je náhodně vygenerován vstupní vektor IV-1.

2. Hlavní klíč (K-2). Klíč je náhodně generovaný spolu se vstupním vektorem IV-2, tyto dvě hodnoty jsou zašifrovány algoritmem AES 256 klíčem K-1 a vstupním vektorem IV -1. Klíč K-2 a vstupní vektor IV -2 slouží k zašifrování dat souboru. [56]

Tabulka 12 Přehled šifrovacích algoritmů aplikace AES Crypt.

Algoritmus	Délka klíčového prostoru.
AES 256 (CBC)	2^{256} $(1{,}15\times10^{77})$
SHA-256	2^{256} $(1{,}15\times10^{77})$

6.1.7.2 Výkonový test (Hashcat benchmark)



Graf 6 Porovnání výsledků výkonových testů podle grafické karty.

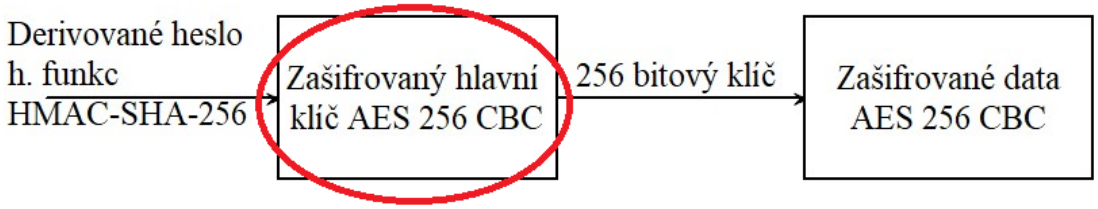
Tabulka 13 Počet dní potřebných k vykonání útoku hrubou silou (GT 1070) v závislosti na délce hesla, znakové sady a použité derivační funkci (v grafu 6).

Znaková sada / Délka hesla	**26**	**36**	**52**	**62**	**96**
4	0	0	0	0	0

5	0	0	3	7	66
6	2	17	161	463	6389
7	65	639	8391	28744	613350
8	1704	23026	436351	1782147	58881639

6.1.7.3 Místo testování

Z výsledků grafu 6 výše můžeme usoudit (v porovnání s rychlostí výkonu AES), že nejslabším místem toho šifrovacího schématu je klíč derivovaný z uživatelského hesla (s označením K-2). [56]



Obrázek 28 Diagram znázorňující využití kryptografických metod k zabezpečení dat pomocí aplikace AES Crypt se zvýrazněným místem testování.

Testování kontejneru softwaru AES Crypt nebylo provedeno, protože nebyl nalezen vhodný nástroj pro jeho otestování.

7 ZHODNOCENÍ VÝSLEDKŮ TESTŮ A DOPORUČENÍ

7.1 AxCrypt

Šifrovací schéma tohoto softwaru používá standardizované metody, které jsou dnes považovány za bezpečné. Data jsou zašifrované pomocí algoritmu AES (128 a 256) CTR, prověřenou a standardizovanou metodou. Klíč k těmto datům je generován náhodně a možnost útoku hrubou silou je tedy značně nepraktická (viz AES-128 výkonový test). Tento klíč je zašifrován v hlavičce souborového formátu pomocí standardizované balící funkce a klíč k těmto datům je derivován z uživatelského hesla funkcí PBKDF s hašovací funkcí SHA-512 (viz výkonový test pod názvem AxCrypt graf 2). Při porovnání těchto dvou rychlostí výkonových testů je zranitelnějším místem uživatelské heslo, z kterého je derivován klíč k hlavičce. Doporučením je zde použití silného hesla (nejlépe náhodně vygenerovaného) s velikostí minimálně 10 znaků a použití delší znakové sady (malé a velké písmena, speciální znaky a číslice).

7.2 VeraCrypt

Tento šifrovací software používá velmi silné, standardizované kryptografické metody spolu se silným šifrovacím schématem. Podobně jako je tomu u schématu AxCrypt, klíč k datům je generován náhodně (za použití tahů myši) a tento klíč zabezpečen v zašifrované hlavičce oddílu klíčem, který je derivován z uživatelského hesla pomocí PBKDF a uživatelem definovaného nastavení (Počet iterací a hašovací funkce). Z grafu 3 je zřejmé, že i s výchozím nastavením je rychlost hledání značně zpomalena. Tento software disponuje komplexnějším nastavením, které značně zvyšuje odolnost proti klíč hledajícím útokům, využitím výkonově náročných kryptografických operací a funkcí. Útok na slabé heslo je zde možný stejně jako u softwaru AxCrypt a to potvrdilo testování kontejneru VeraCrypt, kde bylo heslo se značnou pomocí (zmenšením klíčového prostoru) nalezeno během velmi krátké doby. Podle tabulky 7 je vhodné doporučit délku hesla o minimální délce 9 znacích se znakovou sadou o velikosti 36 a více. Dále je zde nastavení počtu iterací, kde lze navýšením docílit k zpomalení rychlosti útoků, tak i rychlosti odemykání zašifrovaného oddílu. Z grafu 3 je vidět jako nejbezpečnější volbou hašovací funkce Whirlpool, jejíž rychlost v H/s je nejmenší ze všech 4 testovaných funkcí.

7.3 Symantec Encryption Desktop

Šifrovací software Encryption Desktop využívá algoritmus AES-256 k zašifrování dat i všech 3 klíčů ve svém šifrovacím schématu. K derivaci klíče z hesla využívá OpenPGP standardizovanou metodu S2K s hašovací funkcí SHA-1 (graf 4), která dopadla ze všech testovaných funkcí nejhůře. K nalezení slabého hesla, při testování kontejneru, došlo po 21 hodinách mask útoku s rychlostí 2 529 klíčů za sekundu (obrázek 20). Tento software nabízí autentizaci uživatele pomocí bezpečnostního tokenu i ochranu pomocí TPM, jenž jsou velmi silnými prvky ochrany dat a využívají silnou asymetrickou šifru (RSA a Diffe-Hellman o velikosti 1024-4096 bitů). Při vhodném využití těchto ochranných prvků je útok na symetrické klíče znemožněn, avšak odcizení zařízení společně s bezpečnostním tokenem je stále možné. Na základě tabulky 9 je doporučeno využívat silné heslo o délce aspoň 12 znaků a délce znakové sady 52 znaků.

7.4 Microsoft Bitlocker Drive Encryption

Microsoft Bitlocker používá k zabezpečení dat šifrovaného oddílu symetrickou šifru AES se základním nastavení v 256 bitovém XTS módu, jenž je vhodnou standardizovanou metodou pro šifrování dat na disku. K derivaci přístupového klíče z uživatelova hesla využívá hašovací funkci SHA-256, jenž byla otestována v grafu 5 a je znázorněna v algoritmu Bitcracker (obrázek 22). Při testování slabého hesla kontejneru pomocí mask útoku bylo heslo nalezeno po téměř 8 hodinách s rychlostí hledání klíče 237 klíčů za sekundu. Stejně jak Symantec Encryption Desktop i Bitlocker využívá asymetrickou šifru v podobě TPM, smart card a bezpečnostních tokenů. Při kombinaci těchto ochranných prvků k autentizaci a přihlášení uživatele, je útok hrubou silou prakticky znemožněn. Při tvorbě uživatelského hesla je zde důležité používat silné heslo o délce aspoň 10 znaků a delší znakovou sadu obsahující speciální znaky, čísla a velká písmena.

7.5 AES Crypt

AES Crypt je ve srovnání s ostatním softwarem velmi jednoduchý a slouží pouze k šifrování jednotlivých souborů. K šifrování používá jako ostatní prověřenou symetrickou šifru AES s délkou klíče 256 bitů. Data jsou zašifrovány náhodně generovaným klíčem a tento klíč je zašifrován klíčem, jenž je derivován z hesla uživatele hašovací funkcí HMAC-SHA-256 (v grafu 6 nebližší podobná funkce). Z výsledků výkonových testů v grafu 6 a

tabulky 13, je zde doporučeno, aby heslo mělo nejméně 8 znaků v kombinaci se znakovou sadou obsahující i velké písmena, speciální znaky a čísla.

7.6 Porovnání šifrovacích softwarů

Výše zmíněný šifrovací software, využívá kryptografické metody, které se jeví jako dostatečně odolné proti útokům hrubé síly a jejich variantám, avšak za předpokladu, že jejich uživatel využívá dostatečně silné heslo. V této oblasti lze označit, metody aplikované softwarem VeraCrypt, jako nejsilnější v kombinaci s nastavením počtu iterací a hašovací funkce Whirlpool. Problém nedostatečně silného hesla lze omezit využitím dodatečných ochranných metod (TPM, smart card, bezpečnostní token), které podporuje software Bitlocker a Encryption Destkop. Doporučením autora této práce je využití kombinace testovaného softwaru pro diskové šifrování (s TPM a bezpečnostním tokenem nebo smart card) a souborového šifrování.

Tabulka 14 Porovnání šifrovacího softwaru.

	TPM	Bezpečnostní Token	Smart Card	Skrytý oddíl	Celodiskové šifrování	Souborové šifrování
AxCrypt	Ne	Ne	Ne	Ne	Ne	Ano
VeraCrypt	Ne	Ne	Ne	Ano	Ano	Ne
Symantec E. D.	Ano	Ano	Ano	Ne	Ano	Ano
M. Bitlocker	Ano	Ano	Ano	Ne	Ano	Ne
AES Crypt	Ne	Ne	Ne	Ne	Ne	Ano

ZÁVĚR

Jak už autor v této práci uvedl, šifrovací algoritmy jsou pouze jádrem kryptografických metod pro šifrování dat. Neméně významnou součástí je jejich implementace v šifrovacím schématu daného softwaru a schopnost uživatele tento nástroj použít způsobem, který nesnižuje jeho účinnost odolat kryptoanalytickým útokům. Síla hesla bývá často důležitým faktorem při ochraně dat i u softwaru, jehož kryptografické metody jsou v souladu s moderními standardy a doporučeními. Vzhledem k relativně rychlému vývoji technologií je také výpočetní výkon důležitým činitelem odolnosti těchto metod a z velké části také udává dobu, po kterou je vhodné tyto metody užívat. Velký výpočetní výkon poskytují například FGPA a sestavy pro těžení kryptoměny, jež nejsou běžně dostupné kvůli jejich ceně. V první části této práce jsou zde kromě šifer a jejich operačních módu popsány standardy a doporučení, které jsou klíčové z hlediska správného použití těchto základních prvků a poskytují vývojářům softwaru i běžným uživatelům detailní informace o jejich stavu a aktuální bezpečnosti.

V druhé části této práce byl proveden výběr šifrovacího softwaru z dat technicky zaměřených webových magazínů a stránek, které jsou často prvním zdrojem při hledání vhodné ochrany dat nejen pro jednotlivé uživatele, ale i pro malé i velké podniky, které vzhledem k většímu riziku často vyžadují větší míru ochrany. Útoky na klíč zašifrovaných dat, jsou často praktickým ukazatelem bezpečnosti použité metody a pro tento účel je dostupná ke stažení řada programů specializující se ve forenzních metodách obnovy hesla a například i analýzy obrazu dat paměti. Klíče k zašifrovaným datům jsou často odvozeny z uživatelova hesla derivačními algoritmy (PBKDF, S2K) využívající hašovací funkce. Parametry a výpočetní složitost těchto derivačních algoritmů a hašovacích funkcí je kritickým bodem, v šifrovacím schématu využívající heslo zadávané uživatelem, právě kvůli důmyslným útokům soustřeďujícím se na lidskou stránku tohoto problému. Tento problém částečně odstraňují asymetrické šifry, které se využívají k zašifrování symetrických klíčů a poskytují více ochrany, avšak nevýhodou těchto klíčů je jejich délka, způsob jejich přenosu a uskladňování (Bezpečnostní token).

Ve volbě šifrovacího softwaru hraje také zásadní hodnota chráněných aktiv, které se snaží zabezpečit proti možnému odcizení nebo jejich modifikaci. Pokud má tento software chránit kritická data podniku, měl by se jeho způsob a úroveň ochrany dat významně odvíjet od doporučení a nařízení souvisejících s hodnotou chráněných aktiv.

Po porovnání výsledků (v kapitole Výběr kryptografických metod a aplikací pro jejich praktické otestování.) s výsledky provedených testů (v kapitole Porovnání šifrovacích softwarů) a softwarem poskytované ochrany, lze označit Microsoft Bitlocker, jako šifrovací software s nejsilnější kombinací kryptografických metod v testované skupině, avšak vzhledem k tomu, že se nachází pouze na operačním systému Windows, je jeho oblast nasazení značně omezena.

SEZNAM POUŽITÉ LITERATURY

[1] PAAR, Christof a Jan. PELZL. *Understanding cryptography: a textbook for students and practitioners*. New York: Springer, c2010. ISBN 978-3-642-04100-6.

[2] *Introduction to AES Encryption* [online]. In: . Olympia, W, USA: Townsend Security [cit. 2018-01-18]. Dostupné z: https://townsendsecurity.com/sites/default/files/AES_Introduction.pdf

[3] SOSNOWSKI, Rafal. Bitlocker: AES-XTS (new encryption type). *Dubai Security Blog* [online]. [cit. 2018-01-19]. Dostupné z: https://blogs.technet.microsoft.com/dubaisec/2016/03/04/bitlocker-aes-xts-new-encryption-type/

[4] GÓMEZ-RODRÍGUEZ, Carlos. *The Advanced Encryption Standard and its modes of operation* [online]. Coruña, Spain, 2010 [cit. 2018-01-19]. Dostupné z: https://www.maplesoft.com/applications/view.aspx?SID=6618&view=html. Universidade da Coruña.

[5] HOLBREICH, Alexander. Symmetric-key cryptography. In: *Alexander Holbreich* [online]. 2016 [cit. 2018-01-19]. Dostupné z: http://alexander.holbreich.org/symmetric-key-cryptography/

[6] SCHNEIER, Bruce. *Twofish: A 128-Bit Block Cipher* [online]. CA, USA, 1998 [cit. 2018-01-20]. Dostupné z: https://www.schneier.com/academic/paperfiles/paper-twofish-paper.pdf. University of California Berkeley.

[7] SCHNEIER, Bruce. Products that Use Twofish. *Schneier* [online]. [cit. 2018-01-20]. Dostupné z: https://www.schneier.com/academic/twofish/products.html

[8] ANDERSON, Ross. [online]. CA, USA, 2000 [cit. 2018-01-20]. Dostupné z: http://www.cl.cam.ac.uk/~rja14/Papers/serpentcase.pdf. University of California Berkeley.

[9] ANDERSON, Ross. SERPENT. *Univerzita Cambridge: Department of Computer Science and Technology* [online]. UK [cit. 2018-01-20]. Dostupné z: http://www.cl.cam.ac.uk/~rja14/serpent.html

[10] MILANOV, Evgeny. *The RSA Algorithm* [online]. Washington, 2009 [cit. 2018-01-21]. Dostupné z:

https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf. University of Washington.

[11] MANKU, Saikumar a K. VASANTH. BLOWFISH ENCRYPTION ALGORITHM FOR INFORMATION SECURITY. *ARPN Journal* [online]. 2015, (10) [cit. 2018-01-23]. ISSN 1819-6608. Dostupné z: http://cs.indstate.edu/~schinta/blowfish.pdf

[12] KUMAR CHINTA, Sankeeth. *Blowfish* [online]. 2015, Sept 18, 2015, (991730264) [cit. 2018-01-23]. Dostupné z: http://cs.indstate.edu/~schinta/blowfish.pdf

[13] DOLMATOV, ED., V. *GOST R 34.12-2015: Block Cipher "Kuznyechik"* [online]. In: . Research Computer Center MSU, 2016 [cit. 2018-02-05]. ISSN 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc7801#section-2

[14] ALTAWY, Riham a Amr M. YOUSSEF. *A Meet in the Middle Attack on Reduced Round Kuznyechik*[online]. Concordia Institute for Information Systems Engineering, 2015 [cit. 2018-02-05]. Dostupné z: https://eprint.iacr.org/2015/096.pdf. Concordia University.

[15] SCHNEIER, Bruce a John KELSEY. *Related-Key Cryptanalysis of 3-WAY, Biham-DES,CAST, DES-X, NewDES, RC2, and TEA* [online]. Counterpane Systems, U.C. Berkeley [cit. 2018-02-05]. Dostupné z: https://people.eecs.berkeley.edu/~daw/papers/keysched-icics97.ps. U.C. Berkeley.

[16] ADAMS, C. The CAST-128 Encryption Algorithm. In: *RFC* [online]. Entrust Technologies, 1997 [cit. 2018-02-05]. Dostupné z: https://tools.ietf.org/html/rfc2144

[17] HOFFMAN, NICK. *A SIMPLIFIED IDEA ALGORITHM* [online]. Northern Kentucky [cit. 2018-02-05]. Dostupné z: https://www.nku.edu/~christensen/simplified%20IDEA%20algorithm.pdf. Department of Mathematics, Northern Kentucky University.

[18] CHANG, How-Shen. *International Data Encryption Algorithm* [online]. 2004 [cit. 2018-02-05]. Dostupné z: https://users.cs.jmu.edu/abzugcx/Public/Student-Produced-Term-Projects/Cryptology-2002-SPRING/IDEA-by-How-Shen-Chang-2004-FALL.doc. Stamford University Bangladesh.

[19] BONEH, Dan. *Twenty Years of Attacks on the RSA Cryptosystem* [online]. Stanford, CA, 1999 [cit. 2018-02-05]. Dostupné z: https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf. Stanford University.

[20] SCHLAFFER, Martin. *Cryptanalysis of MD4*[online]. Graz, Austria, 2006 [cit. 2018-02-11]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.3624&rep=rep1&type=pdf. Diplomová práce. Graz University of Technology. Vedoucí práce Dipl.-Ing. Dr. techn. Elisabeth Oswald.

[21] STALLINGS, William. The Whirlpool Secure Hash Function. In: *Cryptologia* [online]. 2006, **30**(1), s. 55-67 [cit. 2018-02-11]. DOI: 10.1080/01611190500380090. ISSN 0161-1194. Dostupné z: http://www.tandfonline.com/doi/abs/10.1080/01611190500380090

[22] BIRYUKOV, Alex. *Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1* [online]. Luxembourg, 2016 [cit. 2018-02-11]. Dostupné z: https://eprint.iacr.org/2016/071.pdf. University of Luxembourg.

[23] DOLMATOV, ED., V. *GOST R 34.11-2012: Hash Function: Request for Comments: 6986* [online]. Cryptocom, 2013 [cit. 2018-02-11]. ISSN 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc6986

[24] DOLMATOV, ED., V. *GOST R 34.12-2015: Block Cipher "Kuznyechik": Request for Comments: 7801* [online]. Research Computer Center MSU, 2016 [cit. 2018-02-11]. ISSN 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc7801

[25] *Secure Hash Standard (SHS)*, 2015. In: . NIST Computer Security Division publications, FIPS 180-4. Dostupné také z: https://csrc.nist.gov/csrc/media/publications/fips/180/4/final/documents/fips180-4-draft-aug2014.pdf

[26] *ADVANCED ENCRYPTION STANDARD (AES): Processing Standards Publication 197*, 2001. In: . NIST Computer Security Division publications, FIPS PUB 197. Dostupné také z: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[27] CALLAS, J., 2007. *OpenPGP Message Format: Request for Comments: 4880* [online]. PGP Corporation [cit. 2018-02-11]. Dostupné z: https://tools.ietf.org/html/rfc4880

[28] W. LUCAS., Michael, 2006. *PGP & GPG* [online]. San Francisco: No Starch Press [cit. 2018-02-11]. ISBN 1-59327-071-2. Dostupné z: https://doc.lagout.org/security/No%20Starch%20Press%20-%20PGP%20and%20GPG%20Email%20for%20the%20Practical%20Paranoid.pdf

[29] Security, *Axcrypt* [online]. [cit. 2018-02-27]. Dostupné z: https://www.axcrypt.net/information/security/

[30] *AxCrypt Version 2 Algorithms and File Format*, 2016. Dostupné také z: http://www.axcrypt.net/wp-content/uploads/dlm_uploads/2016/06/AxCryptVersion2AlgorithmsandFileFormat.pdf

[31] *Folder Lock 7 Guide*, Dostupné také z: http://www.newsoftwares.net/doc/folder-lock-7-guide.pdf

[32] *Documentation: Encryption Scheme*, 2017. Dostupné také z: https://www.veracrypt.fr/en/Documentation.html

[33] *PCMAG: Encryption* [online], 2017. New York: ZiffDavis [cit. 2018-03-18]. Dostupné z: https://www.pcmag.com/business/directory/encryption

[34] FEARN, Nicholas, 2018. Top 5 best encryption software tools of 2018. *Techradar.pro: It insights for business.* [online]. Londýn [cit. 2018-03-18]. Dostupné z: https://www.techradar.com/news/top-5-best-encryption-tools

[35] SHARMA, Rishi, 2018. 9 Best Encryption Software For Windows 2018. In: *SYSTWEAK* [online]. Jaipur, Rajasthan [cit. 2018-03-18]. Dostupné z: https://blogs.systweak.com/2017/09/9-best-encryption-software-for-windows/

[36] HUNT, Cale, 2018. Best Encryption Software of 2018. In: *Windows Central* [online]. Mobile Nations [cit. 2018-03-18]. Dostupné z: https://www.windowscentral.com/best-encryption-software

[37] SINGH, Karanpreet, 2018. Top 15+ Best Encryption Software For Windows 2018. *TechViral* [online]. [cit. 2018-03-18]. Dostupné z: https://techviral.net/best-encryption-software-for-windows

[38] ROBB, Drew, 2018. Top 10 Enterprise Encryption Products. *ESecurity Planet* [online]. QuinStreet [cit. 2018-03-18]. Dostupné z: https://www.esecurityplanet.com/products/top-encryption-products.html

[39] *Windows 7 BitLocker™ Drive Encryption Security Policy: For FIPS 140-2 Validation*, 2011. Gaithersburg: NIST.

[40] AGOSTINI, Elena, 2017. *BITCRACKER: BITLOCKER MEETS GPUS* [online]. In: . Řím: DeepSec [cit. 2018-03-19]. Dostupné z: http://technodocbox.com/Windows/71358855-Bitlocker-meets-gpus-bitcracker.html

[41] *Symantec™ Encryption Desktop for Windows: User's Guide*, 2016. Dostupné také z: https://support.symantec.com/content/unifiedweb/en_US/article.DOC9226.html

[42] SMITH, Ryan, 2014. NVIDIA Launches Tesla K80, GK210 GPU. *AnandTech* [online]. [cit. 2018-03-20]. Dostupné z: https://www.anandtech.com/show/8729/nvidia-launches-tesla-k80-gk210-gpu

[43] KUMAR ARYA, Prashant, 2015. *Comparative Study of Asymmetric Key Cryptographic Algorithms* [online]. **5**(1) [cit. 2018-03-22]. ISSN 2249-5789. Dostupné z: http://ijcscn.com/Documents/Volumes/vol5issue1/ijcscn2015050103.pdf

[44] A Symmetric Key Cryptographic Algorithm, 2010. *International Journal of Computer Applications* [online]. **1**(15) [cit. 2018-03-23]. ISSN 0975 - 8887. Dostupné z: http://www.ijcaonline.org/journal/number15/pxc387502.pdf

[45] KNOPF, Christian, 2007. *Cryptographic Hash Functions* [online]. Hannover [cit. 2018-03-23]. Dostupné z: https://www.thi.uni-hannover.de/fileadmin/forschung/arbeiten/knopf-da.pdf. Diplomová práce. Univerzita Hannover.

[46] ROGAWAY, Phillip, 2011. *Evaluation of Some Blockcipher Modes of Operation*. Kalifornie: Kalifornská Univerzita. Dostupné také z: http://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf

[47] WAGNER, David a Helger LIPMAA, CTR-Mode Encryption: Comments to NIST concerning AES Modes of Operations:. In: *Semantic Scholar* [online]. [cit. 2018-03-24]. Dostupné z: https://pdfs.semanticscholar.org/854f/ad4fa592661292ab10ba9add2d890195efb5.pdf

[48] NIST, *Recommendation for Block Cipher Modes of Operation:: The CMAC Mode for Authentication*, 2005. Computer Security Division: NIST. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-38b.pdf

[49] DANG, Quynh, 2008. *Recommendation for Applications Using Approved Hash Algorithms*. Gaithersburg: NIST. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.198-1.pdf

[50] DWORKIN, Morris, 2007. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. Gaithersburg: NIST. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

[51] DWORKIN, Morris, 2004. *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*. Gaithersburg: NIST. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-38c.pdf

[52] *PGP Software Developer's Kit (SDK) Cryptographic Module: FIPS 140-2 Security Policy*, 2008. 3.1.3. Gaithersburg. Dostupné také z: https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp1049.pdf

[53] NIST, *Recommendation for Password-Based Key Derivation: Part 1: Storage Applications*, 2010. Gaithersburg: NIST. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf

[54] *PKCS #5 v2.1: Password-Based Cryptography Standard: RSA Laboratories*, 2012. USA. Dostupné také z: https://www.emc.com/collateral/white-papers/h11302-pkcs5v2-1-password-based-cryptography-standard-wp.pdf

[55] ASUS Radeon RX 460 STRIX Gaming 4 GB review, 2016. *The Guru of 3D* [online]. [cit. 2018-03-26]. Dostupné z: http://www.guru3d.com/articles-pages/asus-radeon-rx-460-strix-gaming-4gb-review,27.html

[56] JONES, Paul, 2016. AES Crypt key generation. In: *Packetizer Forums* [online]. Packetizer [cit. 2018-04-15]. Dostupné z: https://forums.packetizer.com/viewtopic.php?f=72&t=247&p=695&hilit=key+generation#p695

[57] MCALISTER, M., 2010. *PGP® Desktop: Enterprise Whole Disk Encryption Only Edition* . Menlo Park, CA 94025 USA. 08-1622-R-0004. Dostupné také z:

https://www.commoncriteriaportal.org/files/epfiles/pgp-desktop-v9100-sec-eng.pdf

[58] About SciEngines, 2018. *SciEngines* [online]. Kiel [cit. 2018-04-15]. Dostupné z: https://www.sciengines.com/company/about-sciengines/

[59] MATSUI, M., 2004. *A Description of the Camellia Encryption Algorithm*. Network Working Group. Request for Comments: 3713. Dostupné také z: https://tools.ietf.org/html/rfc3713

[60] NFORMATION-TECHNOLOGY PROMOTION AGENCY, *Analysis of Camellia*, 2001. Japonsko. Dostupné také z: https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1082_camellia.pdf

[61] Information technology - Security techniques: Encryption algorithms, *International Organization for Standardization* [online]. Ženeva [cit. 2018-04-16]. Dostupné z: https://www.iso.org/obp/ui/#iso:std:iso-iec:18033:-3:ed-2:v1:en

[62] SCHAAD, J., 2002. *Advanced Encryption Standard (AES) Key Wrap Algorithm*. RSA Laboratories: Network Working Group. Dostupné také z: https://tools.ietf.org/html/rfc3394

[63] DWORKIN, Morris, 2012. *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*. Gaithersburg: National Institute of Standards and Technology. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf

[64] *Hashcat: Advanced password recovery* [online], [cit. 2018-04-22]. Dostupné z: https://hashcat.net/hashcat/

[65] BSI, *Cryptographic Mechanisms: Recommendations and Key Lengths: BSI – Technical Guideline*, 2018. BSI TR-02102-1. Dostupné také z: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=7

[66] *Heureka* [online], 2007. Česko: Heureka Shopping s.r.o [cit. 2018-04-22]. Dostupné z: https://www.heureka.cz/

[67] MITCHELSON, David, 2008. Gainward GTX 1070 Phoenix GS Review. In: *Vortez* [online]. Spojené Království [cit. 2018-04-22]. Dostupné z:

https://www.vortez.net/articles_pages/gainward_gtx_1070_phoenix_gs_review,1.html

[68] HINUM, Klaus, 2005. *NotebookCheck* [online]. Vienna, Austria: Notebookcheck Publishing [cit. 2018-04-22]. Dostupné z: https://www.notebookcheck.net/NVIDIA-GeForce-GT-540M.41715.0.html

[69] AHMED, Shakil et al., 2015. Advanced Encryption Standard-XTS implementation in field programmable gate array hardware. In: *Security and Communication Networks* [online]. **8**(3), s. 516-522 [cit. 2018-04-22]. DOI: 10.1002/sec.999. ISSN 19390114. Dostupné z: http://doi.wiley.com/10.1002/sec.999

[70] YIANNIS, Chrysanthou, 2013. *Modern Password Cracking: A hands-on approach to creating an optimised and versatile attack.* [online]. Londýn [cit. 2018-04-29]. Dostupné z: https://www.ma.rhul.ac.uk/static/techrep/2013/MA-2013-07.pdf. Diplomová práce. Royal Holloway.

[71] SMEJKAL, Miroslav, 2015. *Forenzní analýzy šifrovaných dat*. Zlín. Diplomová práce. Univerzita Tomáše Bati. Vedoucí práce Ing. David Malaník Ph.D.

[72] PRENEEL, Bart a Hans DOBBERTIN, 1997. *The Cryptographic Hash Function RIPEMD-160* [online]. RSA Laboratories [cit. 2018-05-04]. Dostupné z: https://www.esat.kuleuven.be/cosic/publications/article-317.pdf. Vědecký výzkum. Katholieke Universiteit Leuven.

[73] *Passware: Passware Kit Forensic 2017 v5* [online], 1998. Mountain View, CA: Passware [cit. 2018-05-04]. Dostupné z: https://www.passware.com/kit-forensic/

[74] Intel® Xeon® Processor E5-2630 v3, *Intel* [online]. Santa Clara, CA: Intel [cit. 2018-05-05]. Dostupné z: https://ark.intel.com/products/83356/Intel-Xeon-Processor-E5-2630-v3-20M-Cache-2_40-GHz

[75] *NIST: Catalog of Standards* [online], 2012. Gaithersburg, MD: National Institute of Standards and Technology [cit. 2018-05-05]. Dostupné z: https://www.nist.gov/programs-projects/smart-grid-national-coordination/catalog-standards

[76] HUNT, Troy. *';--have i been pwned?: Check if you have an account that has been compromised in a data breach* [online]. haveibeenpwned, 2013 [cit. 2018-05-11]. Dostupné z: https://haveibeenpwned.com/

[77] Brute Force Time Table, 2005. In: *Https://www.reddit.com* [online]. San Francisco, CA: Reddit [cit. 2018-05-11]. Dostupné z: http://i.imgur.com/gfYw57t.png

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instruction
DES	Data Encryption Standard
ČÁST	Šifra pojmenována po tvůrcích Carlisle Adam a Stafford Tavares.
CCM	Counter with Cipher Block Chaining Message Authentication Code
CMAC	Cipher-based Message Authentication code
CU	Compute Unit
CUDA	Compute Unified Device Architecture
FPGA	Field-Programmable Gate Array
FIPS	Federal Information Processing Standards
GB	Giga-Byte
GCN	Graphic Core Next
GDDR5	Graphic Doublle Data Rate 5
GMAC	Galois Message Authentication Code
GOST	Ruský Federální Standard
GPU	Graphic Processing Unit
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IEC	International Electrotechnical Comission
IPSec	Internet Protoc Security
ISO	International Organization for Standardization
Wi-Fi	Wireless Fidelity
SSH	Secure Shell
MD	Merkle Damgard

MDS	Maximum Distance Separable
MHz	Jednotka Mega-Hertz
NBS	National Bureau of Standards
NESSIE	New European Schemes for Signatures, Integrity and Encryption
NIST	National Institute of Standards and Technology
nm	nano-metr
NSA	National Security Agency
OpenCL	Open Computing Language
PES	Proposed Encryption Standard
PHT	Pseudo Hadamard Transforms
PGP	Pretty Good Privacy
PHT	Pseudo-Hadamard Transform
PBKDF	Password Based Key Derivation Function
TB	Tera-Byte
TFLOP	Tera-Floating-point Operation Per Second
TLS	Transport Layer Security
RFC	Request for Comments
SHA	Secure Hash Algorithm
S2K	String to Key
S-Box	Substitution-Box
XTS	Metrická jednotka, nanometr
XOR	XEX-based tweaked-codebook mode with cipher text stealing

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM GRAFŮ

SEZNAM PŘÍLOH

PŘÍLOHA P I: BRUTE-FORCE TABULKA

Entropy (in bits) Entropy = $\log_2(S^L)$ L is pass length S is symbol pool *(i.e. if your password has lowercase and numbers, your **symbol pool** is 26+10=36)* *(If entropy confuses you, just focus on the rest of the chart)*	Length of password approximately equivalent to a given entropy. Note: First three columns are rounded to a max of +/- 0.2. Fourth column is rounded to a max of +/- 0.05. Lowercase (26 symbols, 4.7 bits ea)	UPPER + lower + 0-9 (62 symbols, 5.95 bits ea)	UPPER + lower + 0-9 + special characters (94 symbols, 6.55 bits ea)	Pass **phrase** with an average of **4.3** letters per word (12.93 bits per word) Column's value = words in phrase	Time until *guaranteed* brute-force password crack. Based on attacker's guesses per second vs. password strength. **Formula**: (*Seconds to guaranteed crack*) = ($2^{(entropy)}$) ÷ (guesses per second). Result then converted from *seconds* to more reasonable units of time such as years. **Note**: By definition, it takes **half** of the *guaranteed* crack time on *average* to crack a password. Attacker's brute force guesses per second: 10,000	1,000,000	1,000,000,000	100,000,000,000	1,000,000,000,000	100,000,000,000,000	Entropy (in bits) Entropy = $\log_2(S^L)$ L is pass length S is symbol pool *(i.e. if your password has lowercase and numbers, your **symbol pool** is 26+10=36)* *(If entropy confuses you, just focus on the rest of the chart)*
120		20			∞	∞	∞	421 quadrillion years	42 quadrillion years	421 trillion years	**120**
118	25		18		∞	∞	∞	105 quadrillion years	11 quadrillion years	105 trillion years	118
116				9	∞	∞	∞	26 quadrillion years	2.6 quadrillion years	26 trillion years	116
114		19			∞	∞	658 quadrillion years	6.6 quadrillion years	658 trillion years	6.6 trillion years	114
112	24		17		∞	∞	165 quadrillion years	1.6 quadrillion years	165 trillion years	1.6 trillion years	112
110					∞	∞	41 quadrillion years	411 trillion years	41 trillion years	411 billion years	**110**
108	23	18			∞	∞	10 quadrillion years	103 trillion years	10.3 trillion years	103 billion years	108
106			16		∞	∞	2.6 quadrillion years	26 trillion years	2.6 trillion years	26 billion years	106
104	22			8	∞	643 quadrillion years	643 trillion years	6.4 trillion years	643 billion years	6.4 billion years	104
102		17			∞	161 quadrillion years	161 trillion years	1.6 trillion years	161 billion years	1.6 billion years	102
100					∞	40 quadrillion years	40 trillion years	402 billion years	40 billion years	402 million years	**100**
98	21		15		∞	10 quadrillion years	10 trillion years	100 billion years	10 billion years	100 million years	98
96		16			251 quadrillion years	2.5 quadrillion years	2.5 trillion years	25 billion years	2.5 billion years	25 million years	96
94	20				63 quadrillion years	628 trillion years	628 billion years	6.3 billion years	628 million years	6.3 million years	94
92			14		16 quadrillion years	157 trillion years	157 billion years	1.6 billion years	157 million years	1.6 million years	92
90	19	15		7	3.9 quadrillion years	39 trillion years	39 billion years	392 million years	39 million years	392 millennia	**90**
88					981 trillion years	9.8 trillion years	9.8 billion years	98 million years	9.8 million years	98 millennia	88
86			13		245 trillion years	2.5 trillion years	2.5 billion years	24.5 million years	2.5 million years	25 millennia	86
84	18	14			61 trillion years	613 billion years	613 million years	6.1 million years	613 millennia	6.1 millennia	84
82					15.3 trillion years	153 billion years	153 million years	1.5 million years	153 millennia	1.5 millennia	82
80	17				3.8 trillion years	38 billion years	38 million years	383 millennia	38 millennia	383 years	**80**
78		13	12	6	958 billion years	9.6 billion years	9.6 million years	96 millennia	9.6 millennia	96 years	78
76	16				239 billion years	2.4 billion years	2.4 million years	24 millennia	2.4 millennia	24 years	76
74					60 billion years	599 million years	599 millennia	6 millennia	599 years	6 years	74
72		12	11		15 billion years	150 million years	150 millennia	1.5 millennia	150 years	1.5 years	72
70	15				3.7 billion years	37 million years	37 millennia	374 years	37 years	4.5 months	**70**
68					935 million years	9.4 million years	9.4 millennia	94 years	9 years	34 days	68
66	14	11	10		234 million years	2.3 million years	2.3 millennia	23 years	2.3 years	8.5 days	66
64				5	58 million years	584 millennia	584 years	5.8 years	7 months	2.1 days	64
62	13				14.6 million years	146 millennia	146 years	1.5 years	1.8 months	13 hours	62
60		10	9		3.7 million years	37 millennia	37 years	4.4 months	13 days	3.2 hours	**60**
58					913 millennia	9.1 millennia	9 years	33 days	3.3 days	48 minutes	58
56	12				228 millennia	2.3 millennia	2.3 years	8.3 days	20 hours	12 minutes	56
54		9			57 millennia	571 years	6.8 months	2.1 days	5 hours	3 minutes	54
52	11		8	4	14.3 millennia	143 years	1.7 months	12.5 hours	1.3 hours	45 seconds	52
50					3.6 millennia	36 years	13 days	3.1 hours	19 minutes	11 seconds	**50**
48	10	8			892 years	8.9 years	3.3 days	47 minutes	4.7 minutes	2.8 seconds	48
46			7		223 years	2.2 years	20 hours	12 minutes	1.2 minutes	0.7 seconds	46
44					56 years	6.7 months	4.9 hours	2.9 minutes	18 seconds	0.18 seconds	44
42	9	7			14 years	51 days	1.2 hours	44 seconds	4.4 seconds	0.044 seconds	42
40			6	3.1	3.5 years	13 days	18 minutes	11 seconds	1.1 seconds	0.011 seconds	**40**

Entropy (in bits) Entropy = $\log_2(S^L)$ L is pass length S is symbol pool *(i.e. if your password has lowercase and numbers, your **symbol pool** is 26+10=36)* *(If entropy confuses you, just focus on the rest of the chart)*

Lowercase (26 symbols, 4.7 bits ea) | UPPER + lower + 0-9 (62 symbols, 5.95 bits ea) | UPPER + lower + 0-9 + special characters (94 symbols, 6.55 bits ea) | Pass **phrase** with an average of **4.3** letters per word (12.93 bits per word) Column's value = words in phrase

Length of password approximately equivalent to a given entropy. Note: First three columns are rounded to a max of +/- 0.2. Fourth column is rounded to a max of +/- 0.05.

10,000 | 1,000,000 | 1,000,000,000 | 100,000,000,000 | 1,000,000,000,000 | 100,000,000,000,000

Attacker's brute force guesses per second

Time until *guaranteed* brute-force password crack. Based on attacker's guesses per second vs. password strength. **Formula**: (*Seconds to guaranteed crack*) = ($2^{(entropy)}$) ÷ (guesses per second). Result then converted from *seconds* to more reasonable units of time such as years. **Note**: By definition, it takes **half** of the *guaranteed* crack time on *average* to crack a password.

Entropy (in bits) Entropy = $\log_2(S^L)$ L is pass length S is symbol pool *(i.e. if your password has lowercase and numbers, your **symbol pool** is 26+10=36)* *(If entropy confuses you, just focus on the rest of the chart)*

PŘÍLOHA P 2:PŘEHLED DÉLEK (KLÍČOVÝCH PROSTORŮ) KRYPTOGRAFICKÝCH ALGORITMŮ

Produkt	Algoritmus	Délka klíčového prostoru.
AxCrypt 2	AES 128/256 (CTR)	2^{128} ($3{,}4\times10^{38}$), 2^{256} ($1{,}15\times10^{77}$)
	SHA512	2^{512}($1{,}34\times10^{154}$)
	RSA-4096	2^{4096} ($1{,}04\times10^{1233}$)
Folder Lock	AES 256	2^{256} ($1{,}15\times10^{77}$)
VeraCrypt	AES, Serpent, Twofish, Camellia, Kuznyechik 256 (XTS)	2^{256} ($1{,}15\times10^{77}$)
	RIPEMD-160	2^{160}($1{,}46\times10^{48}$)
	SHA-256	2^{256} ($1{,}15\times10^{77}$)
	Whirpool	2^{512} ($1{,}34\times10^{154}$)
	SHA-512	2^{512}($1{,}34\times10^{154}$)
M. Bitlocker	AES 128/256 (CBC, CCM, XTS)	2^{128} ($3{,}4\times10^{38}$), 2^{256} ($1{,}15\times10^{77}$)
	SHA-256	2^{256} ($1{,}15\times10^{77}$)
	RSA-2048	2^{2048}($3{,}23\times10^{616}$)
AES Crypt	AES 256 (CBC)	2^{256} ($1{,}15\times10^{77}$)
	SHA-256	2^{256}($1{,}15\times10^{77}$)
Symantec E. D.	AES256 (CBC)	2^{256} ($1{,}15\times10^{77}$)
	SHA1	2^{64}($1{,}84\times10^{19}$)
	RSA, Diffe-Hellman	$2^{1024\text{-}4096}$ ($1{,}79\times10^{308}$-$1{,}04\times10^{1233}$)

PŘÍLOHA P3: TABULKA TESTOVACÍHO HARDWARU

Sestava 1. UTB: Server (Ubuntu)	
Grafická karta	Nvidia Tesla K80 24GB
Procesor	2x Intel Xeon E5-2630 v3
Paměti RAM	128GB
Pevný Disk	320GB RAID 1
Sestava 2. UTB: Dekstop (Windows 10)	
Grafická karta	Strix Asus RX460 4GB
Procesor	Intel Core i7-5820K
Paměti RAM	16GB
Pevný Disk	3TB
Sestava 3. Vlastní notebook (Samsung RF511), (Windows 10)	
Grafická karta	GT 540M
Procesor	Intel Core i7-2630QM
Paměti RAM	6GB
Pevný Disk	120GB SSD
Sestava 4. Vlastní desktop (Windows 10)	
Grafická karta	Gainward Pheonix GTX 1070 8GB
Procesor	Intel Core i5-4670K
Paměti RAM	16GB
Pevný Disk	500 a 250 GB SSD