Zeszyty Naukowe Gdańskiej Szkoły Wyższej
2019 ▫ Tom 22 ▫ s. 135–148 ISSN 2300-9055 ▫ Rocznik
DOI 10.24426/zngsw.v22i0.236

Tomasz Muchowski, Szymon Makyła
Wydział Elektryczny, Uniwersytet Morski w Gdyni

Analiza wpływu szyfrowania na operacje odczytu i zapisu

Streszczenie
Szyfrowanie jest kluczowym elementem bezpieczeństwa danych, ponieważ pozwala zabezpieczyć dane przed nieuprawnionym dostępem. Skradziony zaszyfrowany nośnik nie może zostać odczytany, jeżeli napastnik nie zna hasła. Szyfrowanie stanowi więc silny mechanizm zabezpieczający, lecz wiąże się z pewnymi utrudnieniami również dla uprawnionego użytkownika. Do utrudnień tych należą obniżona prędkość operacji odczytu i zapisu, ale także dodatkowe obciążenie procesora wynikające z konieczności szyfrowania danych w locie. Analiza wpływu szyfrowania na wydajność operacji na danych jest istotna, ponieważ pozwala określić teoretyczną użyteczność szyfrowania w zastosowaniu biznesowym. Negatywne konsekwencje zaszyfrowania, w niektórych zastosowaniach, mogą być zbyt wysokie i wiązać się ze niemożnością odczytu lub zapisu danych z wymaganą prędkością. W niniejszej pracy przedstawiono przykład analizy wpływu szyfrowania na operacje odczytu i zapisu dwóch typów pamięci masowych: pamięci flash oraz pamięci talerzowej. Analizie poddano więc pamięci FLASH, dyski SSD oraz dyski HDD. Dysków SSHD nie poddano analizie ze względu na niewielką popularność.

Słowa kluczowe: szyfrowanie, obciążenie procesora, zasoby sprzętowe, odczyt, zapis, test wydajnościowy.

Istota i przedmiot analizy

Przedmiotem analizy jest szyfrowanie i jego wpływ na wydajność pracy urządzenia. Zadaniem szyfrowania jest zabezpieczenie danych kluczem. Proces ten polega na zastosowaniu algorytmu szyfrującego (w postaci funkcji matematycznej) do przekształcenia informacji w postać odczytywalną jedynie po uprzedniej deszyfracji za pomocą algorytmu odwrotnego do algorytmu szyfrującego. Funkcja szyfrująca i deszyfrująca jako parametr przyjmuje klucz szyfrujący, w postaci hasła lub pliku klucza. Zasada działania funkcji szyfrującej może być znana i dostępna publicznie (*open source*), ponieważ wynik działania funkcji oparty jest w pełni na kluczu szyfrującym, natomiast znajomość algorytmu nie umożliwia odwrócenia procesu szyfrowania (Cozzens, 2013; Holden, 2018).

Szyfrowanie stanowi powszechnie znaną metodę zabezpieczania danych, często stosowaną w implementacjach API biznesowych systemów informatycznych, lecz rzadko wykorzystywaną przez administratorów sieci lub użytkowników końcowych. Administratorzy sieci zwykle skupiają się na szyfrowaniu kanałów komunikacji (m.in. poprzez zastosowanie TLS), lecz rzadko na zabezpieczaniu serwerów przed fizycznym wykradnięciem danych. Użytkownicy końcowi natomiast, często obawiają się szyfrowania danych ponieważ uważają, że mogą przez pomyłkę je stracić (Paar and Pelzl, 2010).

Szyfrowanie programowe a szyfrowanie sprzętowe

Pamięci masowe szyfrowane mogą być programowo lub sprzętowo. Szyfrowanie sprzętowe polega na przechowywaniu symetrycznego klucza szyfrującego w pamięci kontrolera dysku, niezależnie od procesora, tym samym eliminując możliwość ataku na pamięć komputera w celu uzyskania klucza. Szyfrowanie programowe obsługiwane jest przez zainstalowane w systemie operacyjnym oprogramowanie, a więc za operacje w pełni odpowiada procesor, zaś klucz deszyfrujący przechowywany jest w pamięci RAM lub pamięci podręcznej procesora (Aumasson, 2017).

Szyfrowaniem programowym objęty może być każdy dysk lub pamięć przenośna, ponieważ działa na poziomie systemu operacyjnego. Najczęściej dane szyfrowane są z wykorzystaniem funkcji systemu operacyjnego (np. BitLocker) lub specjalistycznego oprogramowania (np. VeraCrypt, DmCrypt). Oprogramowanie tworzy na dysku zaszyfrowany wolumin, który nie jest rozpoznawalny przez system operacyjny. Taki wolumin może zostać zamontowany jedynie za pomocą dedykowanego oprogramowania, po podaniu odpowiedniego hasła lub przekazania pliku klucza. Podane hasło jest częścią woluminu — bez niego wolumin nie może zostać prawidłowo zamontowany, a więc i odczytany. Podczas zapisu danych do zaszyfrowanego woluminu dane przekształcane są za pomocą algorytmu szyfrującego przez procesor, a następnie zapisywane są na woluminie. Zawartość woluminu, w tym wykorzystywany system plików zależny jest w pełni od użytkownika. Możliwe jest także zaszyfrowanie partycji systemowej, lecz w takim przypadku konieczne jest zastosowanie specjalnego *bootloadera* (najczęściej dostarczanego przez dostawcę oprogramowania) do odszyfrowania woluminu przed uruchomieniem systemu operacyjnego. Zaszyfrowana w taki sposób partycja systemowa jest dla systemu operacyjnego transparentna i rozumiana jako zwykła partycja. Pomimo tego, podczas pracy systemu, wszystkie dane partycji wciąż są szyfrowane. Szyfrowanie programowe jest więc narzędziem chroniącym dane znajdujące się w woluminie. Wolumin nie jest jednak chroniony przed formatowaniem lub nadpisaniem. W przypadku nadpisania woluminu innym, dane na nim zgromadzone zostają zniszczone (Martin, 2020).

Szyfrowanie sprzętowe wykorzystywane jest głównie w całodyskowym szyfrowaniu (FDE, ang. *Full-Disk Encryption*). W sprzętowym szyfrowaniu wyróżnia się dyski samoszyfrujące oraz samoszyfrujące dyski opieczętowane. Dyski samoszyfrujące (SED, ang. *Self-Encryptiong Drive*) oparte są na standardach OPAL i TCG (Trusted Computing Group, 2020). Kluczem zarządza kontroler dysku, zaś za autoryzację odpowiada procesor (dokonywana jest ona podczas uruchamiania komputera za pomocą specjalnego *bootloadera* lub za pomocą hasła BIOS jeśli wspiera tą opcję płyta główna). Zaletą SED jest pełna transparentność szyfrowania dla systemów operacyjnych oraz przechowywanie, po autoryzacji, klucza w kontrolerze dysku, a nie pamięci wirtualnej komputera. Znacznie zwiększa to bezpieczeństwo, ponieważ nie może on zostać wykradziony poprzez szkodliwe oprogramowanie zainstalowane w systemie operacyjnym. Utrudnione jest także uszkodzenie woluminu, ponieważ kontroler nie umożliwia dostępu do znajdujących się na nim woluminów bez przeprowadzenia procesu autoryzacji. Dane mogą jednak zostać sklonowane lub zniszczone, poprzez fizyczną integrację w kontroler (m.in. *reflash*). Problem ten rozwiązują dyski opieczętowane, ponieważ

wyposażone są w specjalną obudowę, której usunięcie powoduje uszkodzenie talerzy. Dostęp do kontrolera jest więc niemożliwy bez zniszczenia danych.

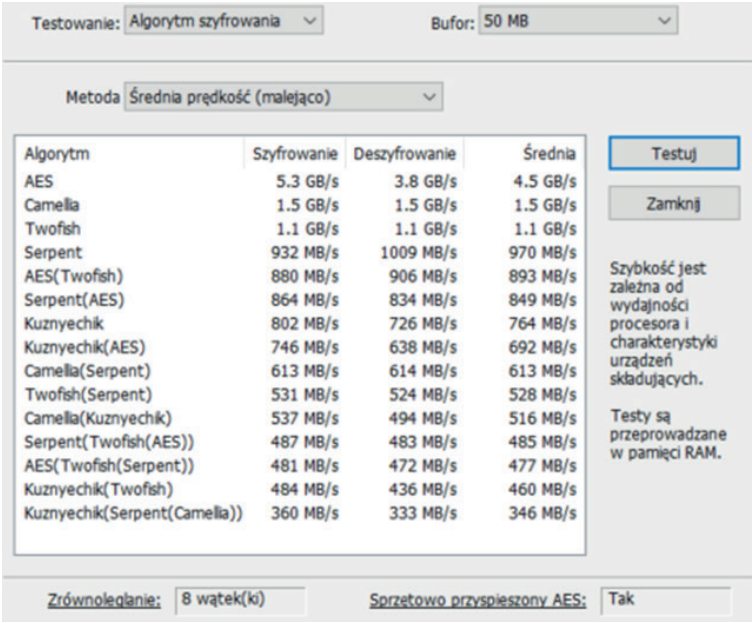
W trakcie analizy zastosowano szyfrowanie programowe, ponieważ celem badania jest wpływ szyfrowania na wydajność operacji odczytu/zapisu. Wpływ ten jest analogiczny dla szyfrowania sprzętowego, ponieważ zadaniem szyfrowania sprzętowego jest wyeliminowanie możliwości naruszenia dysku fizycznie, natomiast proces przetwarzania algorytmu przez procesor się nie zmienia.

Wybór algorytmu szyfrującego

Wpływ na wydajność operacji odczytu/zapisu na zaszyfrowanym urządzeniu ma zastosowany algorytm oraz procesor. W przypadku procesorów opartych na architekturze zawierającej instrukcje szyfrujące wybranego algorytmu szyfrującego, wydajność szyfrowania oraz deszyfrowania znacznie wzrasta (Dobbertin, Rijmen, Sowa, 2005). W trakcie analizy zastosowano algorytm AES oraz procesor wyposażony w zestaw instrukcji AES-NI (Advanced Encryption Standard New Instructions). W skład zestawu wchodzą następujące instrukcje (Intel, 2012a; 2012b):

- AESENC — wykonywanie pojedynczej kolejki cyklu szyfrującego AES,
- AESENCLAST — wykonanie ostatniej kolejki cyklu szyfrującego AES,
- AESDEC — wykonanie pojedynczej kolejki cyklu deszyfrującego AES,
- AESDECLAST — wykonanie ostatniej kolejki cyklu deszyfrującego AES,
- AESKEYGENASSIST — asystowanie w generowaniu kluczy obiegowych AES,
- AESIMC — asystowanie w inwersji kolumn mieszanych AES.

Na rys. 1 przedstawiono wydajność szyfrowania i deszyfrowania danych przez procesor dla popularnych algorytmów.



Algorytm	Szyfrowanie	Deszyfrowanie	Średnia
AES	5.3 GB/s	3.8 GB/s	4.5 GB/s
Camellia	1.5 GB/s	1.5 GB/s	1.5 GB/s
Twofish	1.1 GB/s	1.1 GB/s	1.1 GB/s
Serpent	932 MB/s	1009 MB/s	970 MB/s
AES(Twofish)	880 MB/s	906 MB/s	893 MB/s
Serpent(AES)	864 MB/s	834 MB/s	849 MB/s
Kuznyechik	802 MB/s	726 MB/s	764 MB/s
Kuznyechik(AES)	746 MB/s	638 MB/s	692 MB/s
Camellia(Serpent)	613 MB/s	614 MB/s	613 MB/s
Twofish(Serpent)	531 MB/s	524 MB/s	528 MB/s
Camellia(Kuznyechik)	537 MB/s	494 MB/s	516 MB/s
Serpent(Twofish(AES))	487 MB/s	483 MB/s	485 MB/s
AES(Twofish(Serpent))	481 MB/s	472 MB/s	477 MB/s
Kuznyechik(Twofish)	484 MB/s	436 MB/s	460 MB/s
Kuznyechik(Serpent(Camellia))	360 MB/s	333 MB/s	346 MB/s

Rys. 1. Wydajność operacji szyfrowania i deszyfrowania dla wybranych algorytmów
Źródło: opracowanie własne.

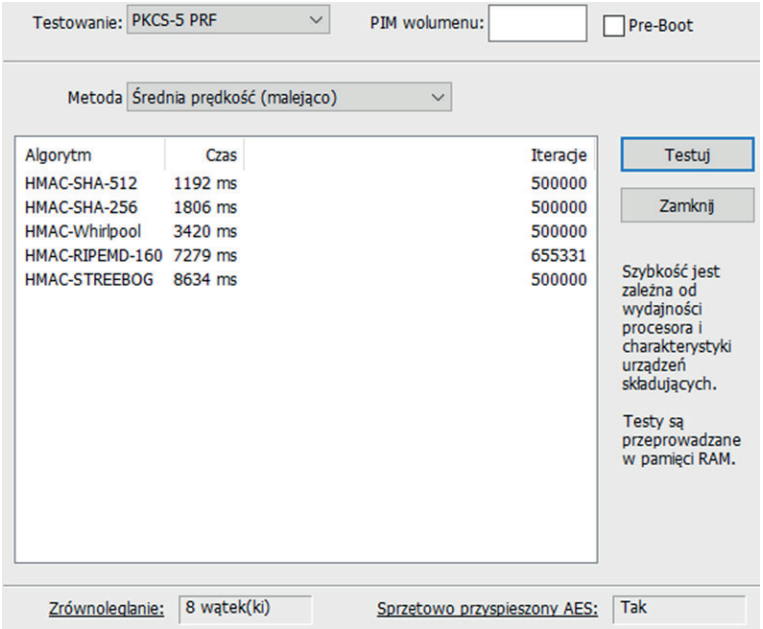
Jak widać, dla wybranego procesora, szyfrowanie i deszyfracja algorytmem AES jest wydajniejsze niż w przypadku pozostałych algorytmów. Wydajność szyfrowania (5,3 GB/s) oraz deszyfrowania (3,8 GB/s) kilkukrotnie przekracza teoretyczną prędkość transferów interfejsu USB 3.1 oraz SATA3, nie przekracza natomiast teoretycznych prędkości transferów interfejsu NVMe. Oznacza to, że wydajność szyfrowania posiadanego procesora bezpośrednio wpływałaby na ograniczenie prędkości transferów dysków NVMe. Należy zauważyć jednak, że testy przeprowadzone zostały w pamięci RAM, a więc pod uwagę nie są brane opóźnienia w dostępie do dysku i wpływ tych opóźnień na szyfrowanie. Przeprowadzony test również nie wykazuje wpływu szyfrowania na działanie pamięci masowej. Stanowi on jednak ważny punkt w wyborze odpowiedniego algorytmu szyfrującego, ponieważ wykazuje, który algorytm pracować będzie najwydajniej na posiadanym sprzęcie komputerowym.

Do poprawnego przeprowadzenia szyfrowania programowego konieczne jest również zastosowanie algorytmu mieszającego (ang. *hashing algorithm*). Zadaniem algorytmu mieszającego jest przekształcenie danych do unikalnej liczby wygenerowanej z ciągu znaków. Algorytm mieszający wykorzystywany jest do stworzenia klucza deszyfrującego wolumin. Klucz deszyfrujący stanowi przetworzone hasło wprowadzone przez użytkownika przy pomocy ciągu znaków stanowiącego podstawę mieszania (ang. *Salt*). W przypadku tworzenia woluminu z wykorzystaniem programu VeraCrypt jako podstawę klucza deszyfrującego brany jest ciąg znaków wygenerowany z sekwencji ruchów myszą (rys. 2) natomiast ilość iteracji mieszania hasła wynosi domyślnie 500 tys.

Rys. 2. Generowanie podstawy mieszania
Źródło: opracowanie własne.

Oznacza to, że hasło zostaje poddane mieszaniu z wykorzystaniem wygenerowanej podstawy, a następnie operacja powtarzana jest określoną ilość razy dla pochodnych mieszania. Celem stosowania dużej ilości iteracji mieszania hasła jest wygenerowanie silnego klucza szyfrującego, którego złamanie metodą *brute force* wymagać będzie dużej ilości czasu. Podczas łamania, każde wygenerowane hasło będzie musiało zostać

poddane 500 tys. iteracji mieszania. Jeżeli napastnik nie zna ilości iteracji mieszania również będzie musiał ją wygenerować. W efekcie złamanie hasła metodą *brute force* staje się kilkuset krotnie trudniejsze, jeżeli napastnik nie posiada informacji o ilości iteracji. Informacje te mogą natomiast zostać pobrane z konfiguracji programu za pomocą wirusa (jeżeli podczas ataku wolumin jest zamontowany oraz jeżeli użytkownik rozkazał oprogramowaniu zapamiętać konfigurację). W praktyce zastosowanie mieszania zwiększa czas uzyskania klucza, co dla napastnika oznacza zwiększony czas łamania hasła, a dla użytkownika zwiększony czas montowania woluminu. Na rys. 3 przedstawiono czas uzyskania klucza deszyfrującego z podanego przez użytkownika hasła dla wybranych algorytmów mieszających.



Rys. 3. Czas uzyskania klucza deszyfrującego z wprowadzonego hasła
Źródło: opracowanie własne.

Jak widać na rys. 3 czas uzyskiwania klucza deszyfrującego dla podanego przez użytkownika (lub napastnika) hasła jest najkrótszy dla algorytmu SHA-512 (Veracrypt, 2020a). Oznacza to, że użytkownik spotka się z czasem montowania woluminu równym 1192 ms, zaś napastnikowi tyle czasu zajmie wygenerowanie i sprawdzenie poprawności pojedynczego hasła (jeśli zna ilość iteracji, jeśli nie będzie musiał hasło przetestować dla całego obsługiwanego przez program zakresu iteracji). Algorytm SHA-512 jest algorytmem najbardziej wydajnym i operację mieszania wykonane zostaną z jego zastosowaniem najszybciej. Użytkownikowi zależeć może jednak na tym aby mieszanie trwało jak najdłużej, aby zmniejszyć prawdopodobieństwo złamania hasła. W takim przypadku użytkownik może wybrać mniej wydajny algorytm, tj. STREEBOG, lecz jest to błędny sposób postępowania. Wydajność algorytmu wysoce zależna jest od sprzętu na którym wykonywana jest operacja mieszania. Napastnik

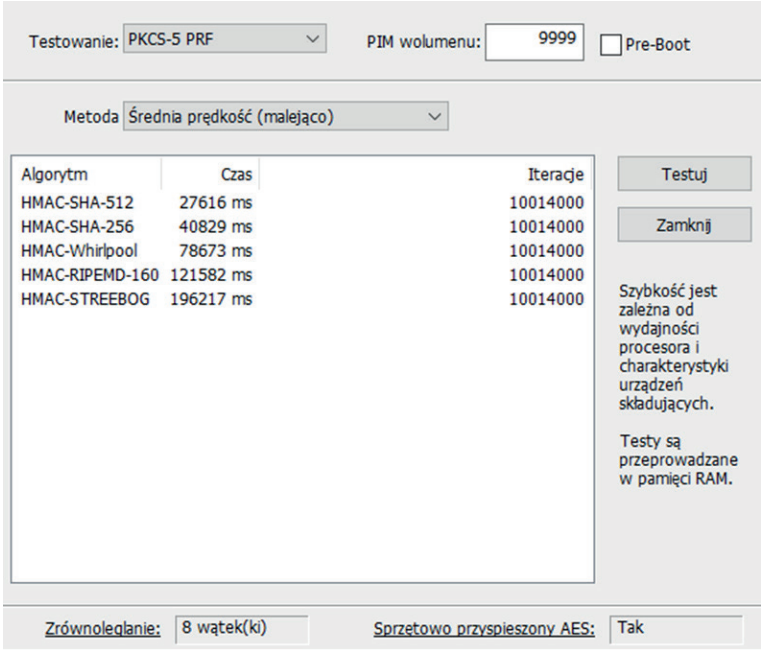
może posługiwać się specjalistycznie skonstruowanym urządzeniem (m.in. łamać hasła za pośrednictwem kart graficznych), które wykonywać będzie szybciej algorytmy inne niż SHA-512. Warto także zauważyć, że algorytm mieszający może zostać złamany, co umożliwi napastnikowi pominięcie narzucanego przez algorytm opóźnienia i przejście do wyniku w czasie niemal błyskawicznym (Elcomsoft Blog, 2020). Z tego powodu warto stosować algorytmy mieszające uznawane za silne i bezpieczne. Algorytm SHA-512 uznawany jest za bardzo bezpieczny, natomiast czas uzyskiwania klucza wydłużyć można zwiększając ilość iteracji. W przypadku oprogramowania VeraCrypt ilość iteracji kontrolowana jest za pomocą parametru PIM (ang. *Personal Iterations Multiplayer*) (Veracrypt, 2020b). Dla algorytmów SHA-512 oraz Whirlpool ilość iteracji wyraża się następującym wzorem:

$$\textit{Ilość iteracji} = 15000 + (\textit{PIM} * 1000). \tag{1}$$

Dla pozostałych algorytmów wykorzystywany jest wzór przedstawiony poniżej:

$$\textit{Ilość iteracji} = \textit{PIM} * 2048. \tag{2}$$

Domyślna wartość PIM dla algorytmów SHA-512 oraz Whirlpool wynosi 485, a dla pozostałych 98. Warto zwiększyć wartość PIM by uzyskać większe bezpieczeństwo. Nie należy jednak stosować zbyt dużej wartości, ponieważ może zbyt mocno przedłużyć czas montowania woluminu (rys. 4) lub wręcz uniemożliwić zamontowanie go.



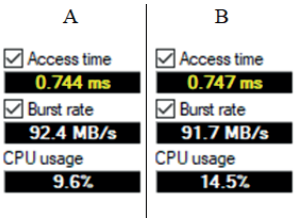
Rys. 4. Czas uzyskiwania klucza deszyfrującego dla dużej wartości PIM
Źródło: opracowanie własne.

Metoda i narzędzia pomiaru wpływu szyfrowania na operacje odczytu i zapisu pamięci masowych

Wpływ szyfrowania na prędkość transferów odczytu i zapisu przetestowano dla dwóch dysków oraz przenośnej pamięci Flash. Zwrócono uwagę na następujące czynniki:
– prędkość transferów odczytu i zapisu,
– ilość wykonywanych w ciągu sekundy operacji wejścia/wyjścia,
– czas dostępu,
– narzut wykorzystania procesora.

Do przeprowadzenia testów wykorzystano programy HD Tune oraz ATTO Disk Benchmark. Oba programy służą do testowania prędkości transferów, lecz HD Tune umożliwia pomiar wykorzystania procesora oraz czasu dostępu. Prędkości transferów wskazywane przez oba programy różnią się, z powodu implementacji innej metody testowania — HD Tune dokonuje odczytu na całej powierzchni pamięci masowej, zaś ATTO Disk Benchmark umieszcza plik testowy na testowanej pamięci masowej, a następnie operuje na wartościach średnich.

Na wszystkich testowanych urządzeniach, każdy brany pod uwagę parametr przetestowano pięć razy w celu wyeliminowania ryzyka wystąpienia anomalii. Jako pierwsze urządzenie testy wykonano na pamięci flash Patriot Rage Elite o pojemności równej 256GB. Stworzona została partycja obejmująca cały obszar pamięci i wykonano testy. Następnie stworzoną partycję zaszyfrowano programem VeraCrypt i ponownie przeprowadzono testy. Testy czasu dostępu oraz wykorzystania procesora wykonano za pomocą programu HD Tune. Wynik przedstawiono na rys 5.



Rys. 5. Porównanie czasu dostępu oraz zużycia procesora dla pamięci flash: A) niezaszyfrowanej; B) zaszyfrowanej
Źródło: opracowanie własne.

Czas dostępu zmienia się nieznacząco i mieści się w granicach błędu. Pozostałe cztery testy nie wykazały zwiększenia czasu dostępu — w przypadku zarówno zaszyfrowanej, jak i niezaszyfrowanej pamięci flash odczyt oscyluje pomiędzy wartościami 0,728 ms i 0,786 ms. Wzrasta natomiast wykorzystanie procesora. Średnie zwiększenie wykorzystania procesora dla wykonanych testów to około 4%.

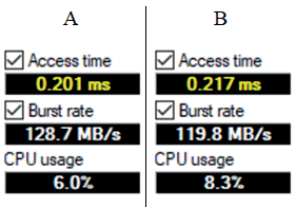
Kolejnym krokiem analizy pamięci flash było wykonanie testów transferów odczytu/zapisu za pomocą programu ATTO Disk Benchmark. Do testów wykorzystano plik o wielkości 256 MB. Wynik testu przedstawiono na rys. 6.

	A				B			
	Write	Read	Write	Read	Write	Read	Write	Read
512 B	188.32 KB/s	7.31 MB/s	375 IO/s	14.96K IO/s	181.16 KB/s	5.97 MB/s	362 IO/s	12.22K IO/s
1 KB	541.52 KB/s	14.82 MB/s	537 IO/s	15K IO/s	501.67 KB/s	11.79 MB/s	501 IO/s	12.07K IO/s
2 KB	1.04 MB/s	28.94 MB/s	484 IO/s	14.64K IO/s	1020.41 KB/s	23.26 MB/s	510 IO/s	11.91K IO/s
4 KB	2.09 MB/s	53.88 MB/s	438 IO/s	13.79K IO/s	1.98 MB/s	44.48 MB/s	506 IO/s	11.39K IO/s
8 KB	3.63 MB/s	66.25 MB/s	455 IO/s	8.40K IO/s	4.01 MB/s	67.23 MB/s	512 IO/s	8.61K IO/s
16 KB	6.18 MB/s	29.69 MB/s	390 IO/s	2.12K IO/s	5.77 MB/s	78.45 MB/s	369 IO/s	5.02K IO/s
32 KB	19.78 MB/s	23.02 MB/s	617 IO/s	1.09K IO/s	13.30 MB/s	36.85 MB/s	425 IO/s	1.18K IO/s
64 KB	28.09 MB/s	68.20 MB/s	493 IO/s	1.08K IO/s	26.60 MB/s	150.59 MB/s	425 IO/s	2.41K IO/s
128 KB	45.29 MB/s	196.26 MB/s	361 IO/s	994 IO/s	51.44 MB/s	271.63 MB/s	411 IO/s	2.17K IO/s
256 KB	50.40 MB/s	307.73 MB/s	346 IO/s	1.20K IO/s	93.70 MB/s	306.25 MB/s	374 IO/s	1.23K IO/s
512 KB	108.47 MB/s	343.62 MB/s	124 IO/s	681 IO/s	94.81 MB/s	302.60 MB/s	189 IO/s	605 IO/s
1 MB	63.84 MB/s	342.86 MB/s	63 IO/s	340 IO/s	72.42 MB/s	305.49 MB/s	72 IO/s	305 IO/s
2 MB	105.57 MB/s	342.09 MB/s	54 IO/s	171 IO/s	91.43 MB/s	306.95 MB/s	45 IO/s	153 IO/s
4 MB	109.87 MB/s	342.09 MB/s	20 IO/s	85 IO/s	95.34 MB/s	307.69 MB/s	23 IO/s	76 IO/s
8 MB	110.11 MB/s	342.09 MB/s	13 IO/s	42 IO/s	96.24 MB/s	309.18 MB/s	12 IO/s	38 IO/s
12 MB	113.30 MB/s	343.60 MB/s	5 IO/s	28 IO/s	93.62 MB/s	311.32 MB/s	7 IO/s	25 IO/s
16 MB	111.55 MB/s	343.62 MB/s	6 IO/s	21 IO/s	79.75 MB/s	306.95 MB/s	4 IO/s	19 IO/s
24 MB	89.80 MB/s	345.85 MB/s	4 IO/s	14 IO/s	93.45 MB/s	306.98 MB/s	3 IO/s	12 IO/s
32 MB	111.30 MB/s	342.86 MB/s	3 IO/s	10 IO/s	79.75 MB/s	307.82 MB/s	2 IO/s	9 IO/s
48 MB	126.32 MB/s	342.18 MB/s	1 IO/s	7 IO/s	63.72 MB/s	311.69 MB/s	1 IO/s	6 IO/s
64 MB	114.46 MB/s	344.62 MB/s	1 IO/s	5 IO/s	95.29 MB/s	304.16 MB/s	1 IO/s	4 IO/s

Rys. 6. Porównanie prędkości transferów odczyt/zapis oraz ilości operacji wejścia/wyjścia dla pamięci flash: A) niezaszyfrowanej; B) zaszyfrowanej
Źródło: opracowanie własne.

Program przeprowadził test dla różnych rozmiarów bloków wejścia/wyjścia (od 512 B do 64 MB). Dzięki temu zauważyć można nie tylko wpływ na prędkości transferów, ale także ilość operacji wejścia/wyjścia możliwych do wykonania w ciągu sekundy dla każdego bloku. Wpływ na prędkość transferów zależy od wielkości bloku wejścia/wyjścia — w przypadku bloków o małym rozmiarze transfery odczytu są wolniejsze o średnio 20%, zaś przy blokach o rozmiarze dużym transfery odczytu są wolniejsze o średnio 10%. Odczyty transferów zapisu wykazują zmniejszenie prędkości w skali od 5 do 10%, lecz na wszystkich przeprowadzonych testach wyniki poszczególnych bloków podlegają nagłym spadkom zarówno w przypadku pamięci niezaszyfrowanej jak i zaszyfrowanej. Z tego powodu pod uwagę brano wynik dla bloku, w którym wartość transferu zapisu była szczytowa dla pamięci zaszyfrowanej. Odczyty ilości operacji wejścia/wyjścia wskazują na obniżenie ilość operacji wykonywanych w trakcie jednej sekundy o 20%.

Jako drugie urządzenie przetestowano dysk SSD Crucial MX300 o pojemności 512 GB. Metodologia testowania jest analogiczna do testowania pamięci flash. Na rys. 7–8 przedstawiono wyniki testów.



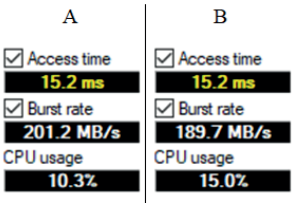
Rys. 7. Porównanie czasu dostępu oraz zużycia procesora dla dysku SSD: A) niezaszyfrowanego; B) zaszyfrowanego
Źródło: opracowanie własne.

	A				B			
	Write	Read	Write	Read	Write	Read	Write	Read
512 B	36.50 MB/s	20.82 MB/s	74.75K IO/s	42.64K IO/s	14.66 MB/s	9.13 MB/s	30.02K IO/s	18.70K IO/s
1 KB	69.62 MB/s	43.73 MB/s	71.29K IO/s	44.78K IO/s	28.59 MB/s	18.90 MB/s	29.28K IO/s	19.35K IO/s
2 KB	136.94 MB/s	99.40 MB/s	70.11K IO/s	50.89K IO/s	54.06 MB/s	39.74 MB/s	27.68K IO/s	20.35K IO/s
4 KB	245.10 MB/s	206.13 MB/s	62.74K IO/s	52.77K IO/s	96.96 MB/s	87.44 MB/s	24.82K IO/s	22.38K IO/s
8 KB	362.06 MB/s	318.50 MB/s	46.34K IO/s	40.77K IO/s	161.94 MB/s	145.22 MB/s	20.73K IO/s	18.59K IO/s
16 KB	429.80 MB/s	409.30 MB/s	27.51K IO/s	26.20K IO/s	243.04 MB/s	224.35 MB/s	15.55K IO/s	14.36K IO/s
32 KB	462.47 MB/s	460.17 MB/s	14.80K IO/s	14.72K IO/s	324.57 MB/s	306.10 MB/s	10.39K IO/s	9.79K IO/s
64 KB	378.71 MB/s	482.80 MB/s	6.06K IO/s	7.72K IO/s	389.88 MB/s	366.88 MB/s	6.24K IO/s	5.87K IO/s
128 KB	428.30 MB/s	486.19 MB/s	3.43K IO/s	3.89K IO/s	419.25 MB/s	410.12 MB/s	3.35K IO/s	3.28K IO/s
256 KB	437.26 MB/s	504.20 MB/s	1.75K IO/s	2.02K IO/s	438.82 MB/s	422.22 MB/s	1.75K IO/s	1.69K IO/s
512 KB	439.48 MB/s	503.19 MB/s	878 IO/s	1.01K IO/s	440.29 MB/s	422.64 MB/s	880 IO/s	845 IO/s
1 MB	423.14 MB/s	503.19 MB/s	423 IO/s	503 IO/s	397.34 MB/s	425.65 MB/s	397 IO/s	425 IO/s
2 MB	432.07 MB/s	503.19 MB/s	216 IO/s	251 IO/s	387.88 MB/s	426.67 MB/s	193 IO/s	213 IO/s
4 MB	431.16 MB/s	501.96 MB/s	107 IO/s	125 IO/s	397.34 MB/s	430.77 MB/s	99 IO/s	107 IO/s
8 MB	426.67 MB/s	504.43 MB/s	53 IO/s	63 IO/s	419.67 MB/s	430.77 MB/s	52 IO/s	53 IO/s
12 MB	428.77 MB/s	502.86 MB/s	35 IO/s	41 IO/s	400.87 MB/s	430.77 MB/s	33 IO/s	35 IO/s
16 MB	436.67 MB/s	504.16 MB/s	27 IO/s	31 IO/s	395.58 MB/s	430.77 MB/s	24 IO/s	26 IO/s
24 MB	429.27 MB/s	505.26 MB/s	17 IO/s	21 IO/s	393.19 MB/s	429.77 MB/s	16 IO/s	17 IO/s
32 MB	441.38 MB/s	503.06 MB/s	13 IO/s	15 IO/s	412.90 MB/s	430.25 MB/s	12 IO/s	13 IO/s
48 MB	431.69 MB/s	497.62 MB/s	8 IO/s	10 IO/s	390.95 MB/s	428.94 MB/s	8 IO/s	8 IO/s
64 MB	433.90 MB/s	497.62 MB/s	6 IO/s	7 IO/s	392.12 MB/s	426.67 MB/s	6 IO/s	6 IO/s

Rys. 8. Porównanie prędkości transferów odczyt/zapis oraz ilości operacji wejścia/wyjścia dla dysku SSD: A) niezaszyfrowanego; B) zaszyfrowanego
Źródło: opracowanie własne.

Również w przypadku dysków SSD czas dostępu nie zmienia się znacząco i mieści się w granicy błędu. Na wykonanych pięciu testach czas ten waha się od 0,201 ms do 0,224 ms zarówno dla zaszyfrowanego, jak i niezaszyfrowanego dysku SSD. Natomiast średni wzrost wykorzystania procesora dla wszystkich wykonanych testów wynosi około 3%. Prędkości transferów odczytu i zapisu dla bloków o małym rozmiarze zostały obniżone o około 40%. Dla bloków o dużym rozmiarze zmniejszenie prędkości transferów odczytu wynosi około 14%, zaś dla zapisu około 10%. Ilość operacji wejścia/wyjścia dla małych bloków zmalała o 40%, a dla dużych bloków o 15%.

Jako trzecie urządzenie przetestowano dysk HDD Seagate Barracuda ST3000DM01 o pojemności 3 TB. Wyniki testów przedstawiono na rys. 9–10.



Rys. 9. Porównanie czasu dostępu oraz zużycia procesora dla dysku HDD: A) niezaszyfrowanego; B) zaszyfrowanego
Źródło: opracowanie własne.

	A				B			
	Write	Read	Write	Read	Write	Read	Write	Read
512 B	13.39 MB/s	13.76 MB/s	27.43K IO/s	28.18K IO/s	4.74 MB/s	4.45 MB/s	9.70K IO/s	9.11K IO/s
1 KB	26.54 MB/s	26.54 MB/s	27.18K IO/s	27.18K IO/s	9.69 MB/s	9.77 MB/s	9.93K IO/s	10K IO/s
2 KB	49.92 MB/s	52.60 MB/s	25.56K IO/s	26.93K IO/s	19.34 MB/s	19.29 MB/s	9.90K IO/s	9.88K IO/s
4 KB	92.60 MB/s	101.58 MB/s	23.71K IO/s	26K IO/s	38.30 MB/s	38.49 MB/s	9.80K IO/s	9.85K IO/s
8 KB	165.82 MB/s	175.51 MB/s	21.22K IO/s	22.46K IO/s	75.16 MB/s	75.34 MB/s	9.62K IO/s	9.64K IO/s
16 KB	205.06 MB/s	206.08 MB/s	13.12K IO/s	13.19K IO/s	99.16 MB/s	101.79 MB/s	6.35K IO/s	6.51K IO/s
32 KB	205.06 MB/s	207.11 MB/s	6.56K IO/s	6.63K IO/s	126.93 MB/s	127.52 MB/s	4.06K IO/s	4.08K IO/s
64 KB	205.57 MB/s	207.11 MB/s	3.29K IO/s	3.31K IO/s	126.82 MB/s	127.38 MB/s	2.03K IO/s	2.04K IO/s
128 KB	204.82 MB/s	202.95 MB/s	1.64K IO/s	1.62K IO/s	128.43 MB/s	130.80 MB/s	1.03K IO/s	1.05K IO/s
256 KB	200.39 MB/s	206.07 MB/s	801 IO/s	824 IO/s	127 MB/s	132.06 MB/s	508 IO/s	528 IO/s
512 KB	201.97 MB/s	209.41 MB/s	403 IO/s	418 IO/s	129.08 MB/s	127.36 MB/s	258 IO/s	254 IO/s
1 MB	208.55 MB/s	208.98 MB/s	208 IO/s	208 IO/s	125.29 MB/s	131.73 MB/s	125 IO/s	131 IO/s
2 MB	207.71 MB/s	198.45 MB/s	103 IO/s	99 IO/s	123.67 MB/s	131.73 MB/s	61 IO/s	65 IO/s
4 MB	201.97 MB/s	206.87 MB/s	50 IO/s	51 IO/s	126.11 MB/s	134.27 MB/s	31 IO/s	33 IO/s
8 MB	198.07 MB/s	206.45 MB/s	24 IO/s	25 IO/s	124.27 MB/s	131.06 MB/s	15 IO/s	16 IO/s
12 MB	201.14 MB/s	211.62 MB/s	16 IO/s	17 IO/s	125.71 MB/s	132.89 MB/s	10 IO/s	11 IO/s
16 MB	208.55 MB/s	212.01 MB/s	13 IO/s	13 IO/s	124.68 MB/s	131.51 MB/s	7 IO/s	8 IO/s
24 MB	201.91 MB/s	207.06 MB/s	8 IO/s	8 IO/s	126.52 MB/s	132 MB/s	5 IO/s	5 IO/s
32 MB	208.13 MB/s	212.01 MB/s	6 IO/s	6 IO/s	128.64 MB/s	128.64 MB/s	4 IO/s	4 IO/s
48 MB	201.40 MB/s	210.99 MB/s	4 IO/s	4 IO/s	121.52 MB/s	128 MB/s	2 IO/s	2 IO/s
64 MB	198.45 MB/s	212.98 MB/s	3 IO/s	3 IO/s	125.08 MB/s	130.83 MB/s	1 IO/s	2 IO/s

Rys. 10. Porównanie prędkości transferów odczyt/zapis oraz ilości operacji wejścia/wyjścia dla dysku HDD: A) niezaszyfrowanego; B) zaszyfrowanego
Źródło: opracowanie własne.

Dla dysków HDD program testowy wykazuje identyczny czas dostępu dla dysku niezaszyfrowanego i dla dysku zaszyfrowanego. Podyktowane może to być faktem zaokrąglenia wyniku do jednego miejsca po przecinku. Średni narzut wykorzystania procesora dla wykonanych testów wynosi około 4,5%. Prędkości transferów odczytu i zapisu dla bloków o małym rozmiarze zostały obniżone o około 70%. Dla bloków o dużym rozmiarze zmniejszenie prędkości transferów wynosi około 40%. Ilość operacji wejścia/wyjścia zmalała dla małych bloków o 70%, a dla dużych bloków o 40%.

Analiza uzyskanych wyników

Wyniki uzyskane z pomiaru pamięci flash, dysku SSD oraz dysku HDD wykazały zwolnienie prędkości transferu po zaszyfrowaniu. Dla każdego typu urządzeń czas dostępu nie zmienia się (utrzymuje się w granicach błędu). Narzut na wykorzystanie procesora mieści się w granicach od 3 do 4,5 procenta. Charakterystyki te są wspólne dla wszystkich przetestowanych urządzeń.

Brak znaczących zmian w czasie dostępu jest zrozumiały, ponieważ szyfrowanie nie wprowadza zmian w systemie plików, natomiast współczesny procesor jest wstanie przeprowadzić operacje deszyfracji wolumenu wystarczająco szybko by nie zakłócić stabilności jego pracy.

Test prędkości transferów odczytu i zapisu bloków o różnych rozmiarach (od 512 B do 64 MB) wykazuje jaką prędkość zapisu i odczytu posiadają pliki o rozmiarach zbliżonych do rozmiarów bloków testowych. Prędkość zapisu i odczytu bloków o małym rozmiarze jest, w przypadku każdego testowanego typu pamięci, znacznie niższa niż bloków o rozmiarze dużym. Spadek ilości operacji wejścia/wyjścia wykonywanych na sekundę jest także znacznie większy dla bloków o rozmiarze małym. Czym większe bloki tym, mniej obniżona jest ilość operacji wejścia/wyjścia wykony-

wanych na sekundę na dysku zaszyfrowanym względem dysku niezaszyfrowanego. Występuje tutaj więc korelacja obniżenia prędkości transferów oraz obniżenia ilości operacji wejścia/wyjścia wykonywanych na sekundę. Założenie, że konieczność przetworzenia dużej ilości kolejek cykli szyfrowania/deszyfrowania przez procesor w przypadku podzielenia pliku testowego na bloki o małym rozmiarze powoduje zbyt duże obciążenie procesora, co prowadzi do opóźnień i obniżenia prędkości transferów, może być trafną tezą. W przypadku dużych bloków plik podzielony jest na znacznie mniej części, co powoduje powstanie mniejszej ilości kolejek cykli szyfrowania/deszyfrowania przez procesor. Wynikiem tego jest zwiększenie prędkości transferów. Zagadnienie to wymaga jednak dalszych badań.

Interesującym faktem jest procentowa różnica spadków prędkości transferów występująca pomiędzy trzema typami urządzeń. Prędkość odczytu bloków o małym rozmiarze na pamięci flash zmalała o 20%, zaś prędkość odczytu bloków o dużym rozmiarze zmalała o 10%. Dla dysku SSD prędkość odczytu bloków o małym rozmiarze i dużym rozmiarze zmalała odpowiednio o 40 i 14%. W przypadku dysku HDD odnotowano spadek prędkości odczytu bloków o małym rozmiarze o około 70%, zaś bloków o dużym rozmiarze o 40%. Różnice te są zastanawiające, ponieważ proces szyfrowania i deszyfracji wykonywany jest programowo. Oznacza to, że jest identyczny dla każdego testowanego urządzenia. Różnice prędkości transferów wynikać muszą więc z budowy urządzeń. Jest to zagadnienie warte uwagi i przeprowadzenia dalszych badań.

Podsumowanie

Szyfrowanie wiąże się ze znacznym spadkiem prędkości transferów pamięci flash, dysków SSD oraz dysków HDD. Występujące różnice budowy wymienionych urządzeń wpływają jednak na stopień obniżenia prędkości. Na stopień obniżenia prędkości wpływa także rozmiar szyfrowanego/deszyfrowanego pliku. Pojedynczy duży plik zostanie zaszyfrowany/rozszyfrowany znacznie szybciej niż duża ilość małych plików.

Prędkość deszyfracji w przypadku pamięci flash zostaje obniżona dla małych plików o około 20%, a dla dużych plików o około 10%. Spadek w przypadku dysku SSD wynosi 40% dla małych plików i 14% dla dużych plików. Konieczność deszyfracji plików dysku HDD obniża prędkość odczytu małych plików o 70%, a dużych o 40%. Szyfrowanie nakłada na prędkość zapisu podobne spadki jak operacja deszyfracji na odczyt.

Przed zaszyfrowaniem dysku warto upewnić się czy jest to właściwe rozwiązanie. Jeśli potrzebny jest częsty dostęp do danych na zaszyfrowanym dysku, należy pamiętać, że szyfrowanie znacznie obniży prędkość operacji odczytu i zapisu, a ponadto powstaje narzut wykorzystania procesora. W niektórych przypadkach może to być nieakceptowalne i spowodować niestabilność systemu. Natomiast jeżeli szyfrowany jest magazyn danych, do których dostęp potrzebny jest sporadycznie (np. dysk z kopiami zapasowymi), obniżenie prędkości oraz narzut wykorzystania procesora jest pomijalny.

Szczególnie wartą uwagi opcją jest szyfrowanie całodyskowe, ponieważ pozwala na zabezpieczenie przed nieuprawnionym dostępem zarówno systemu operacyjnego jak i przechowywanych na dysku danych. Oznacza to, że napastnik nie jest wstanie

odczytać danych systemowych oraz plików konfiguracyjnych i pamięci podręcznej oprogramowania (np. zapisanych haseł w przeglądarce internetowej). W takim przypadku należy jednak zastosować dysk SSD, ponieważ negatywny wpływ szyfrowania na wydajność dysków HDD jest na tyle duży, że może powodować niestabilność pracy systemów (m.in. zawieszanie się programów lub tzw. „lagi"). Szyfrowanie całodyskowe może jednak nie być konieczne. Wystarczyć może zaszyfrowanie jedynie partycji z danymi oraz zmiana nawyków użytkowania systemu operacyjnego oraz oprogramowania na bezpieczniejsze tj. niezapisywanie haseł w przeglądarce, a wykorzystywanie szyfrowanego menadżera haseł oraz wylogowywanie się z usług sieciowych.

Bibliografia

Aumasson, J. (2017). *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press.

Cozzens, M., Miller, S.J. (2013). *The Mathematics of Encryption: An Elementary Introduction (Mathematical World)*. American Mathematical Society.

Dobbertin, H., Rijmen, V., Sowa, A. (2005). Advanced Encryption Standard — AES: 4th International Conference, AES 2004, Bonn, Germany, May 10–12, 2004. *Revised Selected and Invited Papers (Lecture Notes in Computer Science (3373))*. Springer.

Elcomsoft Blog. (2020). *Breaking VeraCrypt*. Pobrane z: https://blog.elcomsoft.com/2020/03/breaking-veracrypt-containers.

Holden, J. (2018). *The Mathematics of Secrets: Cryptography from Caesar Ciphers to Digital Encryption*. Princeton University Press.

Intel. (2012a). *Intel® Advanced Encryption Standard (Intel® AES) Instructions Set — Rev 3.01*. Pobrane z: https://software.intel.com/content/www/us/en/develop/articles/intel-advanced-encryption-standard-aes-instructions-set.html.

Intel. (2012b). *Intel® Advanced Encryption Standard Instructions (AES-NI)*. Pobrane z: https://software.intel.com/content/www/us/en/develop/articles/intel-advanced-encryption-standard-instructions-aes-ni.html.

Martin, K. (2020). *Cryptography: The Key to Digital Security, How It Works, and Why It Matters*. W.W. Norton & Company.

Paar, C., Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer

Trusted Computing Group. (2020). *Self-encrypting Drives (SED) Overview*. Pobrane z: https://trustedcomputinggroup.org/resource/self-encrypting-drives-sed-overview.

Veracrypt. (2020a). *Hash Algorithms*. Pobrane z: https://www.veracrypt.fr/code/VeraCrypt/plain/doc/html/Hash%20Algorithms.html.

Veracrypt. (2020b). *PIM*. Pobrane z: https://www.veracrypt.fr/code/VeraCrypt/plain/doc/html/Personal%20Iterations%20Multiplier%20(PIM).html.

Analysis of the impact of encryption on read and write operations

Abstract
Encryption is a key element of data security as it protects data against unauthorized access. Stolen encrypted media cannot be read if an attacker does not know the password. Encryption is therefore a strong security mechanism, but it also poses some difficulties for an authorized user. These difficulties include reduced speed of read and write operations, but also additional CPU load resulting from the need to encrypt data on the fly. Analysis of the impact of encryption on the performance of data operations is important because it allows to determine the theoretical usefulness of encryption in business applications. The negative consequences of encryption, in some applications, may be too high and may involve the inability to read or write data at the required speed. This paper presents an example of an analysis of the impact of encryption on read and write operations of two types of storage: flash and spinning drives. Thus, FLASH drives, SSDs and HDDs were analyzed. SSHDs were not analyzed due to their low popularity.

Keywords: encryption, CPU load, hardware, read, write, benchmark.