OPEN ACCESS
J. OF ICT, VOL. 5, NO. 1, PP.109- 119, JUNE 2023

ISSN : 2686-1089 (ONLINE)
DOI: 10.52661

Journal of Informatics and Communications Technology (JICT)

Side Channel Attack pada Perangkat Lunak Media Penyimpanan DiskCryptor

Rana Zaini Fathiyana [#1], Siti Zahrotul Fajriyah[#2], Yudiansyah[*3], Rahma Wafii Azahra [#4]

 # Program Studi Teknologi Informasi, Fakultas Informatika, Universitas Telkom
Jalan Halimun Raya No.2, RT.15/RW.6, Guntur, Kecamatan Setiabudi, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12980

** Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom*
Jalan Raya Daan Mogot *No.KM. 11, RW.4, Kedaung Kali Angke, Kecamatan Cengkareng, Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta 11710*

[1] ranazainifathiyana@telkomuniversity.ac.id
[2] sitizahrotul@telkomuniversity.ac.id
[3] yudiansyah@telkomuniversity.ac.id
[4] wafiirahma@student.telkomuniversity.ac.id

Abstract

Meningkatnya popularitas penggunaan perangkat portabel memberikan dampak positif bagi peningkatan produktivitas dan efisiensi, namun menimbulkan dampak lainnya yaitu terjadinya resiko kehilangan data yang signifikan. Perangkat ini dapat dengan mudah hilang atau dicuri. Walaupun harga perangkat portabel tersebut tidak seberapa dibandingkan dengan nilai data yang tersimpan di dalamnya. Teknologi enkripsi merupakan jawaban kebutuhan keamanan data perusahaan, melindungi data di komputer portabel atau media penyimpanan *removable*. *Full disk encryption* (FDE) merupakan solusi untuk menjaga kerahasiaan data yang tersimpan pada perangkat portabel atau laptop. Dengan FDE memungkinkan seluruh kapasitas dan data pada *hard drive* komputer akan diubah menjadi bentuk yang hanya bisa dimengerti oleh orang yang memiliki kunci untuk mendeskripsi data yang telah dienkripsi. DiskCryptor merupakan salah satu contoh perangkat lunak enkripsi yang berjalan pada sistem operasi Windows yang memiliki Lisensi Publik Umum GNU. Pada penelitian ini membahas mengenai ancaman keamanan *(side channel attack)* yang mungkin terjadi di perangkat lunak DiskCryptor diantaranya *Cache Attack, Power Analysis, Timming Attack*, Kebocoran Sistem Operasi, *Password Cracking, Cold-Boot Attack*, dan *Pre-Boot Authentication*.

Keywords: *Full Disk Encryption*, DiskCryptor, Side Channel Attack

I. INTRODUCTION

DEWAN Perwakilan Rakyat Aceh kehilangan sejumlah dokumen penting yang tersimpan di dalam hard disk komputer di ruang komisi yang membidangi hukum, politik, dan pemerintahan tersebut karena dicuri [1]. Pada tanggal 8 Mei 2017, *Covered Entity* (CE), *Bay Area Pain and Wellness Center*, menemukan bahwa mesin *Electromyography* (EMG) yang tersimpan di mobil karyawan telah dicuri. Sebuah laptop yang terpasang pada EMG berisi informasi kesehatan elektronik atau *electronic protected health information* (ePHI) dari sekitar 548 pasien. ePHI berisi data nama pasien dan tanggal lahir. Laptop tersebut telah dilindungi kata sandi namun tidak dienkripsi [2]. Meningkatnya popularitas penggunaan laptop dalam lingkungan perusahaan memberikan

dampak positif bagi peningkatan produktivitas dan efisiensi, namun menimbulkan dampak lainnya yaitu terjadinya resiko kehilangan data yang signifikan. Perangkat ini dapat dengan mudah hilang atau dicuri. Walaupun harga laptop tersebut tidak seberapa dibandingkan dengan nilai data yang tersimpan di dalamnya.

Teknologi enkripsi merupakan jawaban kebutuhan keamanan data perusahaan, melindungi data seperti di laptop atau komputer *desktop*, media penyimpanan *removable*, dan lain-lain. Enkripsi secara signifikan dapat mengurangi atau menghilangkan risiko bisnis yang terkait dengan pelanggaran yang mengakibatkan pengungkapan data rahasia. Enkripsi terbagi menjadi beberapa metode yaitu: *file encryption*, *volume encryption* dan *full disk encryption*.

Full disk encryption (FDE) merupakan solusi untuk menjaga kerahasiaan data yang tersimpan pada perangkat portabel atau laptop. Dengan FDE memungkinkan seluruh kapasitas dan data pada *hard drive* komputer akan diubah menjadi bentuk yang hanya bisa dimengerti oleh orang yang memiliki kunci untuk mendeskripsi data yang telah dienkripsi. Setelah *hard drive* terenkripsi, pengguna perlu *login* ketika komputer pertama kali dinyalakan ini disebut dengan proses *Pre-Boot Authentication*, dapat dengan cara memasukan kata sandi, atau menggunakan token (*smartcard* atau USB). Telah banyak dikembangkan perangkat lunak FDE seperti TrueCrypt, VeraCrypt, BitLocker, DiskCryptor.

Pada penelitian ini akan dibahas mengenai salah satu perangkat lunak FDE yaitu DiskCryptor. DiskCryptor adalah perangkat lunak enkripsi pada *hard drive* yang mempunyai Lisensi Publik Umum GNU untuk sistem operasi Windows. Bahasan difokuskan terhadap kerentanaan di luar aspek kriptografis *(Side Channel)* diantaranya *Cache Attack, Power Analysis, Timming Attack*, Kebocoran Sistem Operasi, *Password Cracking, Cold-Boot Attack*, dan *Pre-Boot Authentication.*

II. LITERATURE REVIEW

A. DiskCryptor

DiskCryptor merupakan sebuah perangkat lunak yang mampu melakukan proses enkripsi terhadap keseluruhan media penyimpanan fisik utuh baik yang sudah terpasang di dalam PC atau *notebook* ataupun yang bersifat portabel (USB, hard disk eksternal). Seluruh data yang tersimpan di dalam media penyimpanan baik file, folder, program, aplikasi, dan sistem operasi akan dienkripsi oleh DiskCryptor.

DiskCryptor menyediakan beberapa fitur enkripsi yaitu mendukung *pre-boot authentication, custom authentication, passphrase strengthening,* dan *two factor authentication.* Serta mendukung enkripsi pada sistem operasi dan *bootable partitions* dengan *pre-boot authentication. Pre-boot authentication* adalah proses mengautentikasi pengguna sebelum komputer boot ke sistem operasi. Sebelum pengguna melihat layar *startup* dari sistem operasi, DiskCryptor akan meminta *username* dan kata sandi saat setelah menghidupkan komputer. Jadi dengan *Pre-boot authentication* komputer akan dalam kondisi boot hingga pengguna *login* dengan *username* dan kata sandi yang benar.

DiskCryptor juga mendukung fitur *pashphrase strengthening* dimana dapat diketahui seberapa kompleks kata sandi yang dibangkitkan oleh pengguna. Adapun tingkatan dari kompleksitas kata sandi yang dibangkitkan adalah *trivally breakable, low, medium, high*, dan yang paling aman yaitu *unbreakable*. Semakin kompleks kata sandi yang dimasukan, semakin tinggi bar *password rating*nya. Lihat indikator pada *password rating* ketika memasukkan kata sandi untuk mengetahui apakah kata sandi sudah cukup kompleks untuk digunakan. Kata sandi dapat berupa abjad (huruf besar atau kecil), numerik, simbol, atau gabungan ketiganya.

Selain pengamanan dengan kata sandi DiskCryptor juga mendukung *custom authentication* dengan penggunaan *keyfiles.* Digunakan untuk meningkatkan keamanan dan bersifat optional. Maksudnya *keyfiles* dapat digunakan bersamaan dengan penggunaan kata sandi dan dapat juga pengamanan dilakukan hanya dengan menggunakan salah satu dari keduanya. *Keyfiles* baik digunakan untuk enkripsi selain enkripsi *system partition* karena ada kemungkinan pengguna tidak dapat masuk kembali ke Windows setelah boot.

Fitur lain yang sangat berguna di DiskCryptor ialah pengguna mampu melakukan jeda ketika proses enkripsi dan dekripsi sedang berlangsung selain itu pengguna dapat melanjutkannya di lain waktu atau bahkan di komputer yang berbeda.

B. Kelebihan dan Kelemahan DiskCryptor

DiskCryptor memiliki kelebihan dan kekurangan seperti pada tabel di bawah ini.

TABLE I
KELEBIHAN DAN KEKURANGAN DISKCRYPTOR

KELEBIHAN	KEKURANGAN
Memiliki kinerja yang baik untuk enkripsi perangkat eksternal maupun internal.	Memiliki *bug*
Mampu melakukan enkripsi lebih dari satu partisi secara bersamaan	Belum banyak *file* bantuan atau dokumentasi
Ukuran unduhan sangat kecil.	
Instalisasi perangkat lunak relatif cepat.	
Mendukung fitur jeda ketika enkripsi untuk melanjutkannya di lain waktu atau bahkan di komputer yang berbeda.	
Mampu mengenkripsi ISO images untuk membuat CD/DVD terenkripsi	
Kompatibel dengan RAID Volume.	
Secara otomatis *dismount* volume ketika *logoff*	
Mampu membuat pintasan keybroard untuk *mounting/dismounting* perangkat dengan cepat.	

C. Side Channel

Keamanan informasi berkaitan erat dengan penerapan ilmu kriptografi, salah satu contoh adalah pemanfaatan aplikasi enkripsi untuk tujuan pengamanan data atau informasi yang menerapkan algoritma kriptografi di dalamnya. Peran penting dari sebuah algoritma kriptografi pada aplikasi kriptografi baik itu berupa *software* ataupun *hardware* menjadi domain utama dalam analisis ataupun penelitian sekarang ini untuk mencari celah keamanan pada algoritma kriptografi tersebut atau yang lebih dikenal dengan istilah *cryptanalysis* (analisis kripto). Namun, analisis kripto sangat sulit dilakukan untuk mendapatkan kunci enkripsi dan membutuhkan waktu yang cukup panjang untuk proses anlisis[3].

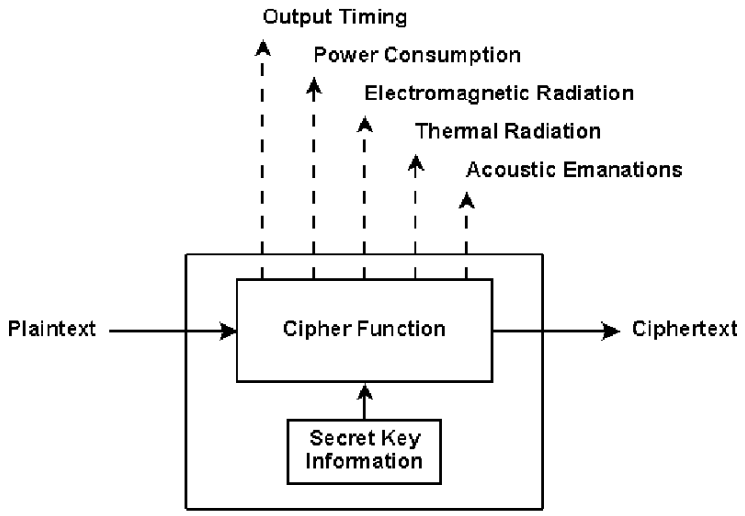

Fig. 1. Indirect *Output* dari Implementasi Blok Cipher *[4]*

Di sisi lain, dalam implementasi algoritma kriptografi ke dalam *software* ataupun *hardware* didapatkan beberapa parameter yang menjadi petunjuk untuk menemukan celah keamanan lain di luar algoritma kriptografi itu sendiri, seperti waktu, konsumsi daya, radiasi elektromagnetik, radiasi termal, sinyal akustik, dan lainnya. Pemanfaatan parameter di luar aspek kriptografi untuk menemukan informasi sensitif atau rahasia mengenai kunci ataupun parameter algoritma lainnya dikenal dengan sebutan *side-channel attack*.

III. RESEARCH METHOD

Metode penelitian yang dilakukan adalah dengan melakukan uji keamanan untuk melihat kerentanan di luar aspek kriptografis *(side channel)* dari perangkat lunak DiskCryptor.

IV. RESULTS AND DISCUSSION

Penggunaan *software-based encryption* tidak lepas dari ancaman-ancaman yang mencoba menembus sistem aplikasi dan mendapatkan informasi sensitif mengenai parameter kriptografi yang digunakan. Diskcryptor sebagai salah satu jenis aplikasi kriptografi yang dikategorikan sebagai *volume encryption* dan *full disk encryption* belum tentu tidak memiliki resiko terhadap keamanan data yang dilindunginya. Vitor dan Marco mengemukakan bahwa terdapat beberapa celah keamanan yang dapat dimanfaatkan oleh penyerang terhadap implementasi dan penggunaan *software-based encryption* [5].

TABLE II
CELAH KEAMANAN PENGGUNAAN SOFTWARE-BASED ENCRYPTION [5]

Encryption Subject	**Vulnerabilities**									
	Plaintext version of old files	Hibernation files	Memory Dump	Paging files	Bad sectors on disks	Brute-force or Dictionary Attacks	Social Engineering	Weak implementation	Support from others	Cold Boot Attack
FE	✓	✓	✓	✓	✓	✓	✓	✓		✓
VDE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FDE					✓	✓	✓	✓	✓	✓

Pada bagian ini, dipaparkan beberapa bentuk ancaman terhadap keamanan penggunaan aplikasi *full disk encryption* seperti Diskcryptor berdasarkan parameter-parameter di luar aspek kriptografi *(side-channel).*

A. Cache Attack

Komputer memiliki beberapa komponen yang berperan penting dalam proses komputasi, seperti prosesor/CPU dan memori. Kecepatan yang dimiliki oleh prosesor/CPU tidak sebanding dengan kecepatan memori yang cenderung lebih lambat, sehingga kecepatan prosesor/CPU memiliki keterbatasan atau limitasi yang diakibatkan lebih lambatnya kecepatan memori. Oleh karena itu, dalam komputer dikenal komponen *cache* dengan kemampuan akses lebih cepat daripada memori dan menjadi jembatan antara CPU dengan memori untuk menjaga kapabilitas CPU dalam kecepatan yang maksimal. *Cache* yang banyak digunakan sekarang ini umumnya terdiri dari 2 (dua) tingkatan, yaitu L1 dan L2/L3. Terdapat dua kondisi pada *cache* ketika blok data diakses, adalah kondisi ***hit*** ketika blok data berada di dalam *cache* dan kondisi ***miss*** ketika blok data tidak berada di dalam ***cache*** dan diambil/dimuat dari memori [6].

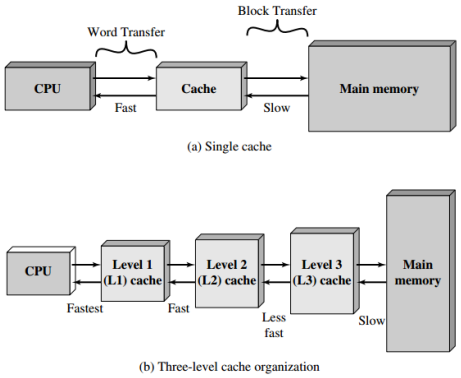

Fig. 2. Cache pada Sistem Komputer
(Sumber: *https://kgptalkie.files.wordpress.com/2016/09/1.png)*

Secara struktur, posisi *cache* berada diantara CPU dan memori yang mengakibatkan informasi dari memori menuju CPU seluruhnya melewati *cache*, termasuk informasi mengenai parameter aplikasi kriptografi. Analisa terhadap penggunaan *cache* untuk menemukan informasi/data tertentu untuk kebutuhan tertentu dikenal dengan sebutan *cache attack*. Bagi seorang penyerang, *cache attack* dapat dimanfaatkan untuk menemukan kunci kriptografi beberapa implementasi algoritma enkripsi, seperti AES, RSA, ECC, dan lainnya. Daniel Gruss mengemukakan dimana *cache attack* dapat dilakukan dalam 2 (dua) metode, yaitu [7]:

a. Flush and reload

Flush and reload memanfaatkan *sharing pages* antara proses milik penyerang dan korban. Dengan *sharing pages*, penyerang dapat memastikan baris memori yang mana yang dikeluarkan dari cache ketika proses dijalankan oleh korban dari *cache* [8]. Metode *flush and reload* memiliki 4 (empat) tahapan yang dilakukan [7] yaitu:

Step 0	penyerang dan korban melakukan pemetaan pada baris di dalam *cache*
Step 1	penyerang melakukan *flush* terhadap baris tersebut
Step 2	korban mengisikan data ke dalam baris tersebut ketika proses enkripsi berjalan
Step 3	penyerang mengambil data (*reload*), jika akses dilakukan dalam waktu yang cepat maka berarti korban mengisikan data, dan sebaliknya jika akses dilakukan dalam waktu lebih lama itu artinya korban tidak mengisikan data ke dalam baris tersebut

Fig. 3. Tahapan *Flush dan Reload*
(Sumber: Gruss, 2016)

Flush and reload dapat dilakukan karena tidak adanya pembatasan terhadap instruksi “clflush“ dalam implementasi perangkat mikroprosesor X86 [8].

b. Prime and probe

Prime and probe merupakan sebuah metode yang dilakukan penyerang untuk mempelajari *cache* mana saja yang diakses oleh korban [9]. Metode *prime and probe* memiliki 3 (tiga) tahapan yang dilakukan [6], yaitu:

Step 0 penyerang mengisikan seluruh baris pada *cache* untuk fungsi monitor (*prime*)
Step 1 korban mengisikan data ke dalam baris yang sudah dimonitor oleh penyerang
Step 2 penyerang melakukan *probe* data untuk menentukan *cache* mana yang diisikan oleh korban

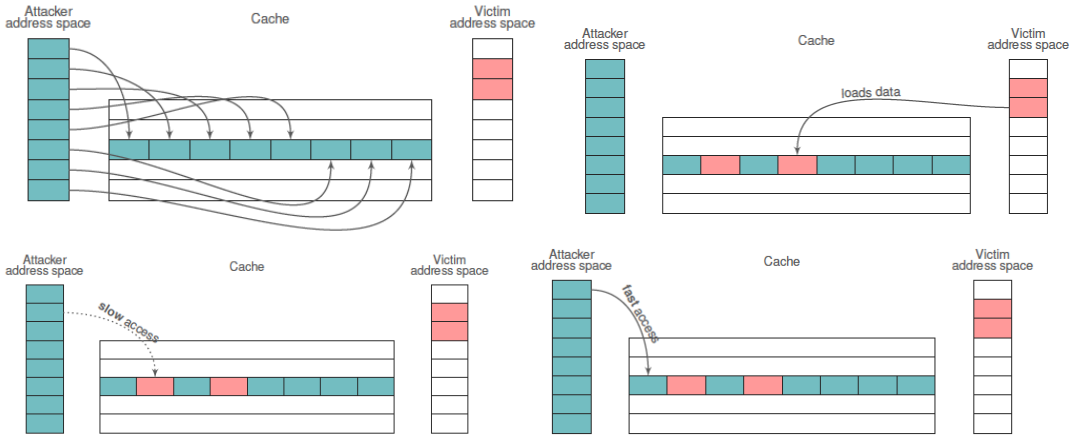

Fig. 4. Tahapan *Prime and Probe [7]*

Dibandingkan dengan *flush and reload*, metode prime and probe menghasilkan lebih banyak *noise* dikarenakan seluruh baris *cache* diisikan dengan data. Gorka dan Xiaofei berhasil mengaplikasikan *cache attack* terhadap beberapa aplikasi/sistem, seperti *browser javascript*, *smartphone application*, AES *key recovery* pada VM, dan infrastruktur *cloud* untuk mendapatkan konten data dari *cache* [10].

B. Power Analysis

Power analysis merupakan sebuah jenis serangan yang dilakukan dengan menganalisa konsumsi daya pada implementasi sebuah algoritma kriptografi [11]. Dengan mempelajari bagaimana konsumsi daya yang diperlukan oleh aplikasi untuk menjalankan fungsi kriptografi (enkripsi dan dekripsi) dapat dilakukan pemetaan untuk menemukan kunci enkripsi yang benar digunakan di dalam memori [12]. Ada beberapa jenis metode dalam melakukan *power analysis*, diantaranya sebagai berikut.

a. Simple Power Analysis

Metode dalam *power analysis* yang menafsirkan secara langsung korelasi dan ketergantungan konsumsi daya dengan instruksi yang sedang dijalankan oleh prosesor (operasi perkalian, pemangkatan/kuadrat, permutasi, pergeseran bit, dan lainnya yang digunakan oleh algoritma kriptografi).

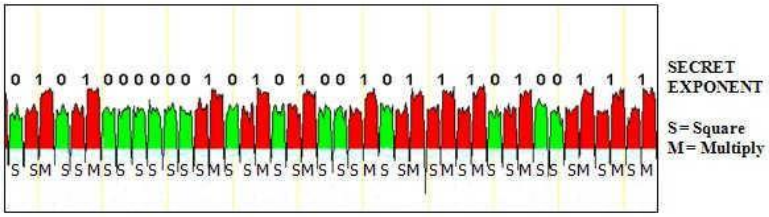

Fig. 5. Konsumsi daya pada operasi kuadrat dan perkalian
(Sumber: https://m.eet.com/media/1109840/Fig1.jpg)

b. Differential power analysis

Metode dalam *power analysis* menggunakan analisa secara statistik dan teknik koreksi terhadap *error* untuk mendapatkan informasi yang berkaitan dengan *secret key* [13]. Kevin Meritt melakukan *differential power analysis* dengan memilih sebuah *byte* dan mengulanginya sebanyak 16 kali untuk sisa *byte* lainnya dan menganalisa konsumsi daya yang digunakan untuk merekonstruksi kunci AES berukuran 128 bit. Diferensial PA memperkecil ruang lingkup analisa kunci AES menjadi hanya 16 x 256 = 4096 kemungkinan yang jauh lebih cepat jika dibandingkan dengan usaha *brute force* sebanyak 2^{128} [4].

c. Correlation power analysis

Correlation power analysis bukan hanya menggunakan analisa statistik, namun juga menggunakan koefisien Pearson untuk korelasi data. *Correlation power analysis* merupakan metode terkini yang digunakan dalam *power analysis* karena memiliki beberapa kelebihan diantaranya tidak perlu melakukan banyak pengambilan data percobaan konsumsi daya (*power trace*) [14].

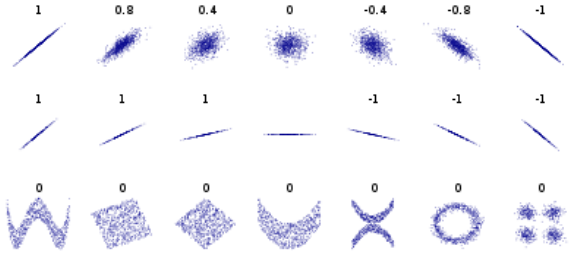

Fig. 6. Koefisien Pearson

C. *Timming Attack*

Timming attack merupakan salah satu teknik serangan *side-channel* dengan menganalisa waktu yang dibutuhkan untuk eksekusi operasi dari implementasi algoritma kriptografi. Proses eksekusi operasi terjadi di dalam *cache* ataupun memori, sehingga informasi mengenai jumlah waktu yang dibutuhkan untuk proses operasi algoritma kriptografi berlangsung dapat dilihat pada *cache* ataupun memori [15]. Informasi mengenai waktu yang dibutuhkan setiap kali operasi enkripsi dapat dianalisa secara statistik untuk menentukan kunci enkripsi yang digunakan [16].

Timming attack dapat diantisipasi menggunakan metode yang ditawarkan oleh Udyani, Janaka, dan Roshan melalui rekayasa waktu yang dibutuhkan untuk proses enkripsi dengan membuat jumlah *clock cycle* menjadi sama jumlahnya dengan *clock cycle* rata-rata pada setiap *round* yang dilakukan [17]. Namun, metode pada [17] mengakibatkan penurunan performa AES yang diimplementasikan. Oleh karena itu, Advait dan Chandane menerapkan pengalian (*multiply*) bilangan 4 (empat) terhadap *clock cycle* untuk setiap *round* [15]. Metode tersebut dapat meningkatkan performa AES dengan tetap mempertahankan ketahanan terhadap *timming attack*.

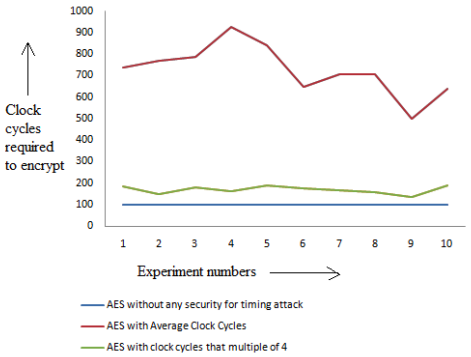

Fig. 7. Grafik performa dari penerapan metode yang dibuat oleh Advait dan Chandane [15]

D. Kebocoran Sistem Operasi

Sebuah sistem operasi memiliki beberapa teknik untuk optimalisasi kecepatan akses dan memberikan kenyamanan bagi penggunanya. Misalkan untuk menghindari aplikasi *crash* akibat sistem kehabisan alokasi memori, sistem operasi seperti Windows menerapkan penggunaan *paging file*. Namun, keunggulan tersebut menjadi kontradiksi jika dilihat dari sudut pandang penyerang yang dapat memanfaatkannya untuk menemukan beberapa informasi penting yang bisa saja dieksploitasi, seperti parameter untuk aplikasi kriptografi.

a. Paging file

Bagian dari aplikasi/program dan data *file* yang tidak cukup disimpan dalam memori, oleh sistem operasi Windows dialihkan ke dalam *swap files* atau *page files* yang berada di dalam *storage*. Sistem operasi menyimpan data-data tersebut yang termasuk data sensitif di dalamnya dalam bentuk *plain* [18]. Kondisi ini tentu membuka informasi mengenai parameter-parameter sensitif yang berkaitan dengan implementasi aplikasi kriptografi, misal seperti parameter kunci kriptografi.

b. Memory dump file

Ketika *error* terjadi dalam sistem operasi (*blue screen*), sistem operasi menuliskan konfigurasi *debug* mengenai memori ke dalam sebuah *memory dump file*. Namun, *memory dump file* juga berisikan data sensitif seperti *cache password*, kunci enkripsi ataupun *file* terbatas yang disimpan tanpa proteksi enkripsi [19].

c. Hibernation file

Ketika komputer dalam kondisi hibernate atau masuk ke dalam mode *power-saving*, data di dalam memori akan dimasukan ke dalam sebuah file *hibernation file* yang berada di dalam storage. *Hibernation file* bisa saja berisikan data sensitif mengenai parameter dari sebuah aplikasi kriptografi.

E. Password Cracking

Proses otentikasi dalam sebuah sistem/aplikasi menjadi garda terdepan yang membatasi akses ke dalam sistem/aplikasi. Umumnya otentikasi diimplementasikan dengan penggunaan *password*. *Password* merupakan rangkaian karakter rahasia yang digunakan untuk membuktikan bahwa benar adalah *user* yang sah yang akan melakukan akses ke dalam sistem/aplikasi. Penyerang menyadari dengan jelas bagaimana *password* menjadi jembatan untuk akses ke dalam sistem/aplikasi, oleh sebab itu tidak sedikit percobaan serangan yang telah banyak dilakukan ditujukan langsung terhadap keamanan *password.* Dalam dunia keamanan informasi, dikenal istilah *password cracking* yang merupakan proses mencoba-coba menebak *password* yang dipakai oleh *user* yang sah untuk mendapatkan akses ke dalam sistem/aplikasi.

Penggunaan *software-based encryption* seperti aplikasi Diskcryptor menggunakan mekanisme otentikasi dengan *password*. *Password* di*input*kan ketika proses *pre-boot* (ketika enkripsi diterapkan pada *volume* yang berisikan sistem operasi) ataupun proses *mounting* (ketika enkripsi diterapkan pada USB atau media penyimpanan eksternal lain). Proses otentikasi tersebut dapat dimanfaatkan penyerang untuk menemukan *password* yang tepat yang dapat dilakukan dalam 3 (tiga) metode sebagai berikut [20].

a. ***Dictionary attack***, metode yang dilakukan dengan menggunakan daftar kata-kata yang umumnya digunakan untuk *password*, jika *password* menggunakan kata yang sederhana dan umum, dictionary attack dapat menemukan *password* tersebut dengan cepat.

b. ***Hybrid attack***, serupa dengan metode dictionary attack hanya saja ada penambahan beberapa angka ataupun simbol sederhana untuk menemukan *password* yang benar.

c. ***Brute force***, mencoba seluruh kemungkinan kombinasi karakter (huruf, angka, symbol, dan sebagainya) untuk menemukan *password* yang benar, hanya saja prosesnya membutuhkan waktu yang cukup lama.

F. Cold-Boot Attack

Memori komputer berisikan informasi keseluruhan aktivitas sistem operasi, seperti *processes, registry keys, event logs, network artefact, hardware and software configuration, malware (rootkit)*, atau bahkan informasi sensitif seperti *user generated content* (*password dan caches*). Ada anggapan bahwa memori komputer akan terhapus isinya secara langsung ketika aliran listrik terputus (*power off*), atau dengan kata lain sulit untuk mengambil kembali data di dalam memori tersebut, namun beberapa peneliti dari Universitas Princeton membuktikan bahwa angggapan tersebut kurang tepat dan menjadi kelemahan bagi penggunaan memori [21]. Berikut merupakan beberapa karakteristik umum perangkat memori, seperti:

a. Setiap bit data disimpan terpisah di dalam kapasitor pada IC,
b. Kondisi/status kapasitor di dalam memori hanya ada 2, yaitu ***charge*** dan ***discharged***,
c. Kondisi ***discharged*** terjadi secara perlahan,
d. Dinamic (perlu secara periodic dilakukan *refresh*),
e. Data akan terhapus secara periodik pada suhu normal (*volatile*).

Keutuhan data di dalam memori sangat dipengaruhi oleh suhu temperatur perangkat. Pada suhu normal 20^0 – 25^0 C dan dalam kurun waktu 6 detik, keutuhan data hilang sampai 51,9%. Dengan menurunkan suhu temperatur pada perangkat memori sampai dengan -50^0 C, keutuhan data hanya hilang sampai dengan 0,18% [37]. *Cold-boot attack* menjadi ancaman bagi keamanan aplikasi Diskcryptor yang digunakan dengan perangkat memori SDRAM, DDR, DDR2, DDR3 dan DDR4 dengan kerawanan bahwa konten yang tersimpan di dalam memori dapat dilakukan rekonstruksi, termasuk data kunci untuk enkripsi. Sebagai bentuk antisipasi *cold boot attack* dan untuk menambah ketahanan perangkat memori yang digunakan untuk aplikasi Diskcryptor,

G. Pre-Boot Authentication

Pada bagian sebelumnya diketahui bahwa ancaman keamanan terhadap aplikasi Diskcryptor salah satunya dapat terjadi akibat adanya akses fisik terhadap memori agar dapat dilakukan *imaging*, rekonstruksi dan identifikasi kunci AES. Di sisi lain, Jonathan Brossard melakukan pendekatan lain yang sama sekali tidak membutuhkan akses fisik untuk mendapatkan *password* (dalam bentuk *plain*) pada memori [22].

Penggunaan aplikasi *full disk encryption* untuk enkripsi media penyimpanan secara umum menggunakan mekanisme *pre-boot authentication* yang dimana sebelum sistem operasi berjalan, terlebih dahulu dilakukan proses otentikasi terhadap *user*.

Fig. 8. *Pre-boot Authentication* Aplikasi Diskcryptor (API)

Pre-boot authentication yang dilakukan sebelum booting sistem operasi berjalan hanya memiliki tampilan layaknya BIOS API seperti pada gambar III.29, hal ini dikarenakan tidak ada kernel di dalam memori ketika proses *pre-boot authentication berlangsung* [22]. *Pre-boot authentication* meminta *input*an dari *user* sebuah *password* untuk otentikasi yang diketikan melalui *keyboard* untuk selanjutnya diproses oleh BIOS *interruption*. Dari metode *input*an *keyboard* (*keystroke*) tersebut, dapat dikenali bagaimana BIOS *interruption* berhubungan dengan *input keystroke* yang disimpan pada BIOS *keyboard buffer*. BIOS *keyboard buffer* berada di dalam BIOS data area pada alamat offset 0x40:0x1e (dengan panjang 32 byte).

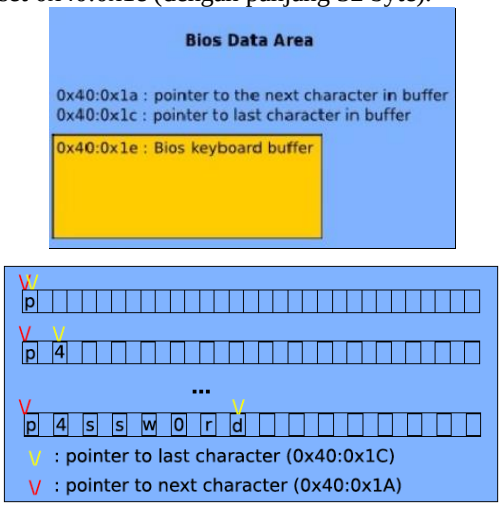

Fig. 9. BIOS Data Area dan Kondisi BIOS Keyboard Buffer ketika *Input Keystroke*

Kondisi BIOS *keyboard buffer* yang sudah diisikan *user* selanjutnya diterima dan diproses oleh BIOS *interruption*, namun isi konten pada BIOS *keyboard buffer* tidak dikosongkan/dihapus. Kondisi seperti ini menjadi celah keamanan dan dapat dimanfaatkan penyerang untuk menemukan *password* yang masih disimpan dalam bentuk *plain* pada BIOS *keyboard buffer*. Walaupun *user* melakukan *input keystroke* berulang kali (lebih dari sekali), tetap saja informasi *password* dapat ditemukan di dalam BIOS *keyboard buffer*. Hal ini dapat terjadi karena karakteristik dari BIOS *keyboard buffer* yang merupakan *rotative buffer*, mengakibatkan informasi *password* pada *buffer* hanya digabungkan dan dipisahkan dengan sebuah karakter 0x1a
Serangan terhadap pre-boot authentication pada aplikasi Diskcryptor dapat dilakukan terhadap versi aplikasi 0.2.6 yang dijalankan pada sistem operasi Windows (model otentikasi lain seperti USB *token, smartcard* ataupun *biometric* berada di luar ruang lingkup bahasan) [22].

V. CONCLUSION

Penggunaan aplikasi DiskCryptor memberikan layanan jaminan keamanan data/konten yang ada di dalam media penyimpanan digital, baik dalam bentuk volume disk ataupun full disk. Namun tidak menutup adanya ancaman terhadap keamanan aplikasi Diskcryptor. Oleh karena itu, untuk meningkatkan ketahanan dalam penggunaan aplikasi Diskcryptor yang dilihat dari sisi pengguna dan juga pengembang aplikasi, disarankan beberapa hal berikut:

a. Menerapkan enkripsi menyeluruh terhadap seluruh volume di dalam komputer (menggunakan opsi enkripsi full disk) guna menghindari adanya kebocoran informasi parameter kriptografi terkait aplikasi yang berada di volume lainnya (*OS volume*).
b. Menggunakan fitur-fitur aplikasi yang berkaitan dengan optimalisasi keamanan, seperti penggunaan *strong password*, *key file* dengan enkripsi, *external bootloader* dan *wipe cache password.*
c. Melakukan optimalisasi kode sumber untuk memperkuat ketahanan aplikasi terhadap serangan *side-channel* dengan teknik *hiding* dan *masking*.

REFERENCES

[1] Bakri, "Usut Tuntas Pencurian Data di Komisi A DPRA", 24 February 2014. [Online]. Available: http://aceh.tribunnews.com/2014/02/24/usut-tuntas-pencurian-data-di-komisi-a-dpra. [Accessed 2 February 2018].
[2] U.S. Departement of Health and Human Services, "Breach Portal: Notice to the Secretary of HHS Breach of Unsecured Protected Health Information," Washington DC, 2017.
[3] Xiaoyu, Z., Kaiyan, C., Yang, Z., Weilong, G., & Lei, L. (2015). Correlation Power Analysis for AES Encryption Device. 4th National Conference on Electrical and Computer Engineering (NCEECE 2015).
[4] Meritt, K. (2012). Differential Power Analysis Attacks on AES.
[5] Vitor Hugo Galhardo, Marco Aurelio A. Henriques, A Comparison of Encryption Tools for Disk Data Storage from Digital Forensic Point of View, University of Campinas: School of Electrical and Computer Engineering (FFEC), 2016
[6] Canteaut, A., Lauradoux, C., & Seznee, A. (2006). Understanding Cache Attacks. INRIA.
[7] Gruss, D. (2016). Cache Side-Channel Attacks and The Case of Rowhammer. IAIK, Graz University of Technology.
[8] Yarom, Y., & Falkner, K. (2014). Flush+Reload: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. The University of Adelaide.
[9] F. Liu, Y. Yarom, Q. Ge, G. Heiser and R. B. Lee, "Last-Level Cache Side-Channel Attacks are Practical," 2015 IEEE Symposium on Security and Privacy, San Jose, CA, 2015, pp. 605-622.
[10] Irazoqui, G., & Guo, X. (2017). Cache Side Channel Attack: Exploitability and Countermeasures. Black Hat Asia 2017.
[11] http://www.iet.unipi.it/g.dini/Teaching/sncs/lectures/handouts/12.side_channels_attacks.pdf
[12] Unterluggauer, T., & Mangard, S. (2016). Exploiting the Physical Disparity: Side-Channel Attacks on Memory Encryption. IAIK, Graz University of Technology.
[13] Kocher, P., Jaffe, J., & Jun, B. (1998). Introduction to Differential Power Analysis and Related Attacks. Cryptography Research.
[14] Gamaarachchi, Hasindu & Ganegoda, Harsha. (2018). Power Analysis Based Side Channel Attack.
[15] Kelapure, A., & Chandane, M. M. (2015). Countermeasures against Timing Attack on AES. International Journal of Computer Science and Information Technologies, Vol. 6 (4), 4106-4109.
[16] Sarma, D. (2012). Security of Hard Disk Encryption. KTH Information and Communication Technology.
[17] Herath, U., Alawatugoda, J., & Ragel, R. (2014). Software Implementation Level Countermeasures Against the Cache Timing Attack on Advanced Encryption Standard.
[18] Andrew Y, "True Crypt Free Open Source On-The Fly Encryption Un-Official. Paging File", [Online]. Available: https://andryou.com/truecrypt/docs/paging-file.php [Accessed 31 Maret 2018].
[19] Andrew Y, "True Crypt Free Open Source On-The Fly Encryption Un-Official. Memory Dump Files", [Online]. Available: https://andryou.com/truecrypt/docs/ memory-dump-files.php [Accessed 31 Maret 2018].
[20] Tasevski, P. (2011). Password Attacks and Generation Strategies. Tartu University.
[21] Halderman, J. A., Schoen, S. D., Paul, W., dkk. (2008). Lest We Remember: Cold Boot Attacks on Encryption Key.
[22] Brossard, J. (2008). Bypassing Pre-boot Authentication Passwords by Instrumenting the BIOS Keyboard Buffer (Practical Low Level Attacks Against x86 Pre-boot Authentication Software).