Second Preimage Attack pada Skema Davies-Meyer Berbasis SIMECK32/64 Menggunakan Metode Kortelainen

Aji Bagas Putranto[1)], Andriani Adi Lestari[2)]
(1) Badan Siber dan Sandi Negara, aji.bagas@bssn.go.id
(2) Politeknik Siber dan Sandi Negara, andriani.adi@poltekssn.ac.id

Abstrak

Second preimage attack metode Kortelainen merupakan suatu skema serangan yang diaplikasikan pada fungsi hash berstruktur Merkle-Damgard. Second preimage attack metode Kortelainen memiliki dua variasi, yaitu Chosen Initial Value Attack (CIVA) dan Chosen Prefix Attack (CPA). Serangan ini memanfaatkan struktur intan untuk mencari second preimage dari suatu pesan. Struktur intan merupakan suatu pohon biner yang tersusun dari nilai-nilai hash yang berkolisi. Salah satu skema fungsi hash yang berstruktur Merkle-Damgard adalah skema Davies-Meyer. Skema Davies-Meyer merupakan skema fungsi hash yang memanfaatkan block cipher sebagai fungsi kompresi dan dikatakan sebagai skema fungsi hash yang aman. Pada Makalah ini dilakukan dua variasi second preimage attack metode Kortelainen pada skema Davies-Meyer berbasis SIMECK32/64. Serangan variasi CIVA dilakukan dengan dua struktur intan ($d = 11$ dan $d = 8$). Variasi serangan ini memperoleh dua second preimage untuk $d = 11$ dan satu second preimage untuk $d = 8$. Kompleksitas waktu serangan untuk $d = 11$ adalah $2^{25} + 2^{20}$ komputasi dan $d = 8$ adalah $3 \cdot 2^{22,5} + 2^{21}$ komputasi. Serangan variasi kedua, yaitu CPA dengan nilai $d = 8$ memperoleh satu second preimage. Kompleksitas serangan variasi kedua ini adalah $2^{23,5} + \sqrt{8} \cdot 2^{21}$.

Kata Kunci: *Chosen Initial Value Attack* (CIVA),*Chosen Prefix Attack* (CPA), Davies-Meyer, *Secod Preimage Attack* Metode Kortelainen, SIMECK32/64, Struktur Intan.

1. PENDAHULUAN

Secara umum, terdapat dua jenis fungsi *hash*, yaitu *dedicated hash function* dan fungsi *hash* berbasis *block cipher* [1]. Fungsi *hash* berbasis *block cipher* dapat dikonstruksi dengan memanfaatkan skema Merkle-Damgard. Merkle [2] dan Damgard [3] menyatakan bahwa suatu fungsi *hash* dapat dikonstruksi dengan fungsi kompresi yang *collision resistant* dan diproses secara sekuensial. Salah satu fungsi kompresi yang dapat digunakan untuk mengonstruksi fungsi *hash* adalah *block cipher* [4]. Akan tetapi, apabila *block cipher* diterapkan secara langsung pada skema Merkle-Damgard, terdapat kerawanan yang dapat dimanfaatkan. Pesan dan *IV* yang digunakan sebagai masukan dapat diperoleh dengan melakukan proses dekripsi sehingga tidak memenuhi salah satu sifat fungsi *hash* yaitu *one-wayness*.

Permasalahan tersebut menjadi latar belakang perlunya modifikasi skema dengan memanfaatkan struktur Merkle-Damgard. Modifikasi yang dilakukan yaitu dengan melakukan operasi XOR antara nilai masukan pada skema (pesan atau *IV*) dengan keluaran dari algoritme *block cipher* yang dilakukan oleh Preneel *et al.* [5] untuk membangun fungsi *hash* dengan memanfaatkan *block cipher*. Langkah ini dilatarbelakangi oleh sulitnya mendesain skema fungsi *hash* yang aman. Skema milik Preneel *et al.* [5] selanjutnya dikenal dengan skema PGV yang memiliki 64 variasi dengan 12 variasi dikatakan aman [6]. Dari 12 variasi PGV, terdapat 3 variasi yang dikatakan ideal. Tiga skema tersebut selanjutnya dikenal dengan skema Davies-Meyer, Matyas-Meyer-Oseas, dan Miyaguchi-Preneel [4].

Suatu fungsi *hash* dikatakan aman jika memenuhi 3 sifat, yaitu *preimage resistance* (*one-wayness)*, *second preimage resistance* (*weak collision resistance*), dan *collision resistance* (*strong collision resistance*) [1]. *Second preimage resistance* merupakan suatu kondisi ketika diberikan nilai masukan x_1 maka akan sulit ditemukan nilai masukan x_2 dengan $x_1 \neq x_2$ sedemikian sehingga nilai *hash* $h(x_1) = h(x_2)$ [1]. Salah satu metode dalam melakukan *second preimage attack* diperkenalkan oleh Andreeva *et al.* [7] pada fungsi *hash* berstruktur Merkle-Damgard dengan memanfaatkan struktur intan.

Struktur intan merupakan suatu struktur data berupa pohon biner. Pembentukan struktur intan dilatarbelakangi oleh proses pencarian kolisi yang dilakukan secara berulang. Proses ini dilakukan dengan mengambil himpunan nilai *hash* berbeda dalam jumlah besar hingga himpunan nilai *hash* tersebut menjadi suatu nilai *hash* tunggal. Hasil kolisi tersebut disusun sedemikian sehingga membentuk struktur pohon biner yang menyerupai struktur intan. Struktur intan pertama kali diperkenalkan oleh Kelsey dan Kohno [8]. Struktur intan direpresentasikan sebagai suatu pohon biner yang terdiri dari himpunan nilai *hash* berbeda sebagai daun (*leaves*) dan nilai *hash* tunggal sebagai akar (*root*). Kompleksitas data yang dibutuhkan untuk mengonstruksi struktur intan Kelsey dan Kohno [8] yaitu $\sqrt{k} \times 2^{\frac{n+k}{2}}$ dan kompleksitas waktu yang dibutuhkan adalah $n \times$

$\sqrt{k} \times 2^{\frac{n+k}{2}}$ dengan nilai n sebagai ukuran keluaran algoritme dan k sebagai panjang blok pesan [9]. Berdasarkan pengembangan ini, Kortelainen dan Kortelainen [10] membuat suatu metode untuk melakukan konstruksi struktur intan dengan kompleksitas yang lebih baik dibandingkan dengan konstruksi struktur intan Kelsey dan Kohno [8]. Metode ini dilakukan dengan memanfaatkan nilai *hash* dan blok pesan yang sebelumnya telah terbentuk.

Struktur intan milik Kortelainen dan Kortelainen [10] selanjutnya digunakan sebagai metode awal untuk melakukan s*econd preimage attack* yang juga dilakukan oleh Kortelainen dan Kortelainen [11]. *Second preimage attack* Kortelainen dan Kortelainen [11] dikonstruksi dengan mengombinasikan struktur intan Kortelainen dan Kortelainen [8] dan *second preimage attack* Andreeva *et al.* [7]. *Second preimage attack* Kortelainen dan Kortelainen [10] memiliki 2 variasi serangan yaitu *Chosen Initial Value Attack* (CIVA) dan *Chosen Prefix Attack* (CPA). Pada variasi pertama yaitu CIVA, penyerang diperbolehkan untuk memilih nilai awal dari fungsi *hash.* Pada variasi kedua yaitu CPA, penyerang mencari *second preimage* dengan memanfaatkan *prefix* yang ada. Kompleksitas dari kedua serangan ini yaitu $O(2^{\frac{n+d}{2}} + \sqrt{d} \cdot 2^{n-d})$ dengan nilai n sebagai ukuran keluaran algoritme dan nilai d sebagai panjang dari struktur intan. Serangan ini diterapkan pada fungsi *hash* yang berstruktur Merkle-Damgard. Andreeva *et al.* [7] mengatakan bahwa konstruksi Merkle-Damgard memiliki beberapa kelemahan dilihat dari beberapa serangan yang telah dilakukan, salah satunya berupa serangan kolisi yang menjadi sarana untuk mencari *second preimage.* Salah satu skema fungsi *hash* yang memiliki struktur Merkle-Damgard yaitu Davies-Meyer.

Berdasarkan uraian di atas, pada penelitian ini akan dilakukan *second preimage attack* dengan metode Kortelainen dan Kortelainen [11] pada skema Davies-Meyer. *Second preimage attack* milik Kortelainen dan Kortelainen [11] yang digunakan pada penelitian ini selanjutnya disebut dengan *second preimage attack* metode Kortelainen. Pemilihan algoritme *block cipher* yang diterapkan pada skema Davies-Meyer didasarkan pada kompleksitas serangan dan keterbatasan komputasi. Berdasarkan hal tersebut, algoritme SIMECK32/64 digunakan sebagai fungsi kompresi pada skema Davies-Meyer. SIMECK32/64 menggunakan masukan berupa nilai $n = 32$ sehingga kompleksitas total yang dibutuhkan untuk melakukan *second preimage attack* adalah 2^{24}. Nilai kompleksitas tersebut memungkinkan serangan dapat dilakukan dengan sumber daya yang tersedia. Selain itu, SIMECK32/64 dikonstruksi dengan mengombinasikan komponen terbaik algoritme SIMON32/64 dan SPECK32/64 yang diklaim lebih efisien dan padat [12].

2. LANDASAN TEORI

2.1. Skema Davies-Meyer

Skema Davies-Meyer merupakan suatu skema fungsi *hash* yang menggunakan *block cipher* sebagai fungsi kompresi E. Skema Davies-Meyer ditunjukkan pada Gambar 1.

Gambar 1 Skema Davies-Meyer [4]

2.2. SIMECK

Pesan yang akan dienkripsi dibagi menjadi 2 *words* l_i dan r_i. Fungsi putaran didefinisikan sebagai

$$R_{k_i}(l_i, r_i) = (r_i \oplus f(l_i) \oplus k_i, l_i),$$

dengan fungsi $f(l_i)$ didefinisikan sebagai

$$f(l_i) = (l_i \odot (l_i \lll 5)) \oplus (l_i \lll 1).$$

Operasi yang digunakan dalam fungsi putaran algoritme SIMECK yaitu:

a. $\oplus$ merupakan fungsi XOR;
b. $\odot$ merupakan fungsi AND;
c. $\lll x$ merupakan rotasi ke kiri sebanyak x bit;
d. k_i merupakan kunci putaran ke-i;
e. l_i merupakan n bit *most significant bit* dari pesan; dan
f. r_i merupakan n bit *least significant bit* dari pesan.

Kunci putaran k_i dibangkitkan dari kunci master K. Pertama, kunci master K dibagi menjadi 4 *words* dan digunakan sebagai nilai awal (t_2, t_1, t_0, k_0) pada *feedback shift register*. Proses pembaruan penjadwalan kunci direpresentasikan sebagai berikut

$$k_{i+1} = t_i,$$
$$t_{i+3} = k_i \oplus f(t_i) \oplus C \oplus (z_j)_i.$$

Indeks i menunjukkan proses putaran ke-i dengan $0 \leq i \leq 32$. Indeks j merujuk pada barisan z_j yang digunakan pada algoritme SIMECK dengan $j = 0,1$. z_0 digunakan pada SIMECK32/64 dan SIMECK48/96. z_1 digunakan pada SIMECK64/128. Dalam proses pembaruan, *least significant bit* dan *most significant bit* dari K yang masing-masing berukuran n bit dimuat dalam k_0 dan t_2. Fungsi putaran dengan konstanta $C \oplus (z_j)_i$ sebagai kunci putaran digunakan kembali untuk melakukan proses pembaruan *register* dan membangkitkan kunci putaran. Nilai dari konstanta yang digunakan yaitu $C = 2^n - 4$.

2.3. Struktur Intan

Kortelainen dan Kortelainen [10] melakukan konstruksi struktur intan dengan membagi proses konstruksi ke dalam *Jump*, *Phase*, dan *Step*.

a. *Jump*

Konstruksi struktur intan D dengan 2^d variabel berantai untuk $d \geq 2$, dilakukan dalam d *Jumps* yaitu $J_d, J_{d-1}, \dots, J_1$ dengan d merupakan tinggi pohon biner. Variabel berantai merupakan n-bit masukan dari langkah sebelumnya pada suatu fungsi *hash* iteratif. Notasi pada *Jump* adalah sebagai berikut.

1) J_i merupakan proses *Jump* ke-i dengan $d \geq i \geq 1$.
2) H_i merupakan himpunan dari 2^d variabel berantai.
3) B_i merupakan himpunan pasangan variabel berantai dan pesan dengan nilai *hash* yang sama.

Proses *Jump* diawali dengan proses *Initialization* yang dijelaskan lebih rinci pada butir d. Pada *Jump* J_i, pasangan variabel berantai dan pesan dengan nilai *hash* yang sama dihimpun dalam himpunan B_i. Anggota himpunan B_i selanjutnya digunakan sebagai masukan pada J_{i-1}. Nilai masukan ini dihimpun dalam H_{i-1} sehingga $H_{i-1} = \{f(h,x) | (h,x) \in B_i\}$ dengan kardinalitas H_{i-1} yaitu 2^{d-1}.

Pada *Jump* J_{i-1}, himpunan pasangan B_{i-1} dibentuk dari himpunan H_{i-1} sehingga dihasilkan himpunan H_{i-2} yang memenuhi $H_{i-2} = \{f(h,x) \,|\, (h,x) \in B_{i-1}\}$ dengan kardinalitas 2^{d-2} dan digunakan sebagai masukan pada *Jump* selanjutnya.

b. *Phase*

Dalam setiap *Jump* J_i, dilakukan *Phase* sebanyak j dengan $i \geq j \geq 1$ yaitu $P_{i,j}, P_{i-1,j-1}, \dots, P_{2,2}, P_{1,1}$. Notasi pada *Phase* adalah sebagai berikut:

1) $P_{i,j}$ merupakan *Phase* ke-j pada proses *Jump* ke-i.
2) K_{i-1} merupakan himpunan bagian dari H_i.
3) T_{i-1} merupakan himpunan pasangan dari himpunan K_{i-1}. Himpunan pasangan merupakan himpunan yang berisi dua nilai awal (IV) dan dua pesan berbeda yang memiliki nilai hash yang sama.
4) $H_i \backslash K_{i-1}$ merupakan himpunan H_i tanpa anggota himpunan K_{i-1}.

Hasil yang didapatkan dari proses *Phase* $P_{i,j}$ yaitu himpunan pasangan T_{i-1} dari K_{i-1}, sehingga kardinalitas T_{i-1} yaitu 2^{i-1}. Selanjutnya, pada *Phase* P_{i-1} dibentuk himpunan pasangan T_{i-2} dari K_{i-2} dengan $K_{i-2} \subseteq H_i \backslash K_{i-1}$. Proses ini dilakukan kembali hingga didapatkan himpunan pasangan T_1 dari sub himpunan $K_1 \subseteq H_i \backslash (K_{i-1} \cup K_{i-2} \cup \dots \cup K_2)$ pada *Phase* $P_{2,2}$. Pada *Phase* ini terbentuk himpunan K_0 dengan dua nilai *hash* sebagai elemen dari himpunan, sehingga pada *Phase* $P_{1,1}$ dilakukan pencarian himpunan pasangan T_0 dari himpunan K_0. Keluaran yang didapatkan adalah himpunan B_i yang memenuhi $B_i = T_{i-1} \cup T_{i-2} \cup \dots \cup T_0$. Selanjutnya, himpunan B_i menjadi himpunan masukan pada *Jump* J_{i-1}. Proses yang dilakukan untuk mencari himpunan pasangan T_{i-1} disebut dengan proses *Step*.

c. *Step*

Setiap *Phase* terdiri dari beberapa *Step*. Setiap proses *Phase* $P_{i,j}$ terdiri dari 2^{j-1} *Step*. Proses *Step* dinotasikan $S_{i,j,k}$ yang berarti proses *Step* ke-k pada *Phase* ke-j dan *Jump* ke-i. Masukan pada setiap *Step* adalah $A_{j,k}, M_{j,k}$, dan $H_{j,k}$. Notasi pada *Step* adalah sebagai berikut:

1) $j \in \{2,3, \dots, d\}$ dan $k \in \{0,1,2, \dots, 2^{j-2} - 1\}$, dengan j, k merupakan indeks proses *Phase* dan *Step*.
2) $A_{j,k}$ merupakan himpunan $2^j - 2k$ nilai *hash*.
3) $h_{j,k}, h'_{j,k} \in A_{j,k}$ merupakan nilai *hash* dalam himpunan $A_{j,k}$.
4) $M_{j,k}$ merupakan himpunan dari blok pesan.
5) $x'_{j,k} \in M_{j,k}$ merupakan blok pesan dalam himpunan $M_{j,k}$.
6) $M'_{j,k}$ merupakan himpunan dari blok pesan baru yang dibangkitkan.
7) $x'_{j,k} \in M'_{j,k}$ merupakan blok pesan baru dalam himpunan $M'_{j,k}$.
8) $H_{j,k}$ merupakan himpunan nilai *hash* dari $(A_{j,k}, M_{j,k})$.

Pada setiap *Step* dilakukan pencarian pasangan nilai *hash* dan blok pesan yang berkolisi. Untuk mencari kolisi tersebut nilai awal $(h_{j,k}, h'_{j,k})$ dibagi ke dalam himpunan $A_{j,k}$ dengan jumlah nilai awal pada masing-masing himpunan $A_{j,k}$ adalah

$$|A_{j,k}| = 2^j - 2k \quad \text{..... (1).}$$

Setiap nilai $h_{j,k}$ pada himpunan $A_{j,k}$ di-*hash* dengan pesan $M_{j,k}$ yang merupakan bagian dari pesan $M'_{j,k}$ yang dibangkitkan. Jumlah pesan $M_{j,k}$ yang diproses di setiap *Step* adalah

$$|M_{j,k}| = \left\lceil \frac{2^{\frac{n+j}{2}-1}}{2^j - 2k} \right\rceil \quad \text{..... (2).}$$

Hasil *hashing* $A_{j,k}$ dengan $M_{j,k}$ dihimpun dalam himpunan $H_{j,k}$. Himpunan $H_{j,k}$ menjadi himpunan nilai *hash* yang akan dibandingkan dengan proses *hashing* kedua. Pada proses *hashing* kedua, himpunan pesan $M'_{j,k}$ dibangkitkan. Banyaknya anggota himpunan pesan $M'_{j,k}$ yang dibangkitkan adalah

$$|M'_{j,k}| = \left\lceil \frac{2^{\frac{n-j}{2}-1}}{2^j - 2k} \right\rceil \quad \text{..... (3).}$$

Proses *hashing* kedua dilakukan dengan melakukan *hashing* antara $A_{j,k}$ dan $M'_{j,k}$ sesuai dengan Persamaan 3. Kemudian dicari $h_{j,k}, h'_{j,k}$ dan x_{j_k}, x'_{j_k} yang berkolisi sedemikian sehingga memenuhi Persamaan 4.

$$f(h_{j,k}, x_{j,k}) = f'(h'_{j,k}, x'_{j,k}) \quad \text{..... (4).}$$

Dalam proses ini diasumsikan hanya terdapat pasangan $(h_{j,k}, x_{j,k})$ dan $(h'_{j,k}, x'_{j,k})$ yang berkolisi.

Dengan demikian, didapatkan keluaran sebagai berikut

1) Himpunan $A_{j,k+1} = A_{j,k}\{h_{j,k}h'_{j_k}\}$;
2) Himpunan $M_{j,k+1} = M_{j,k} \cup M'_{j,k}$;
3) Himpunan $H_{j,k+1} = f\left(A_{j,k+1}, M_{j,k+1}\right)$; dan
4) Himpunan $B_i = B_i \cup \left\{\left(h_{j,k}, x_{j,k}\right), \left(h'_{j,k}, x'_{j,k}\right)\right\}$.

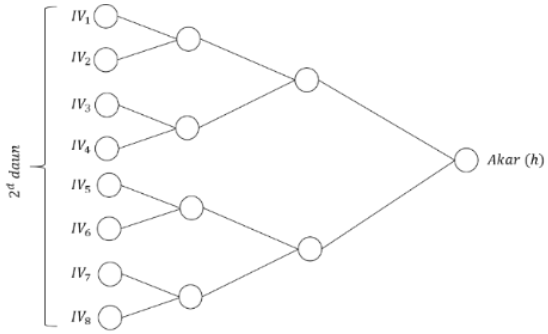
Keluaran dari *Step* $S_{(i,j,2^{j-2}-1)}$ atau *Step* terakhir dari *Phase* $P_{i,j}$ menjadi masukan untuk $S_{i,j-1,0}$ yaitu *Step* pertama dari *Phase* $P_{i,j-1}$ untuk setiap $j \in \{3,4,\dots,d\}$.

d. *Initialization* $I(i)$

Initialization $I(i)$ merupakan proses awal pada *Jump* J_i. Masukan pada proses ini adalah himpunan $A_{i,0}$ yang merupakan bagian dari himpunan H_i berupa 2^d nilai *hash*. Proses dalam tahap $I(i)$ yaitu

1) Membangkitkan himpunan blok pesan $M_{i,0}$ sebanyak $2^{\frac{n-i}{2}-1}$ dengan $i = d$;
2) Menghitung nilai *hash* $H_{i,0} = f(A_{i,0}, M_{i,0})$.

Keluaran yang dihasilkan adalah himpunan $A_{i,0}; M_{i,0}; H_{i,0}$. $A_{i,0}$ merupakan himpunan nilai *hash* yang digunakan sebagai nilai awal pada setiap proses pencarian pasangan kolisi. Skema struktur intan diilustrasikan pada Gambar 2.



Gambar 2 Struktur intan dengan $d = 3$ [10]

2.4. *Second Preimage Attack* Menggunakan Metode Kortelainen

Kortelainen dan Kortelainen [11] mengembangkan *second preimage attack* dengan ide dasar dari pembentukan struktur intan adalah untuk mengambil himpunan nilai *hash* dalam jumlah besar dan mencari nilai *hash* tunggal sebagai hasil dari konvergensi himpunan nilai *hash*. Dua variasi *second preimage attack* yang disebut *Chosen Initial Value Attack* (CIVA) dan *Chosen Prefix Attack* (CPA) dilakukan dengan memanfaatkan nilai *hash* tunggal dari struktur intan. Serangan variasi CIVA memungkinkan penyerang dapat memilih nilai awal (IV) dari fungsi *hash*. Serangan variasi CPA memungkinkan penyerang dapat menentukan *prefix* pada pesan asli dan membuat struktur intan dengan menggunakan *prefix* yang sama. *Second preimage* yang dibentuk dengan variasi kedua harus mengandung *prefix* yang sama dengan *prefix* yang telah ditentukan.

CIVA dan CPA menggunakan fungsi iteratif $f^+: \{0,1\}^n \times (\{0,1\}^m)^+ \to \{0,1\}^n$ dari fungsi kompresi f dengan $m > n$. Notasi $+$ pada fungsi f melambangkan sifat iteratif sementara $(\{0,1\}^m)^+$ melambangkan blok pesan masukan pada fungsi iteratif f. Variasi CIVA memungkinkan penyerang untuk dapat memilih nilai awal dari himpunan bagian $I = \{h_1, h_2, \dots, h_{2^d}\}$ dengan kardinalitas 2^d dan $d < n$. Penyerang memilih nilai awal $h_0 \in I$ dan pesan $x \neq y$ sedemikian sehingga $f^+(h_0, x) = f^+(h_0, y)$. Pada variasi CPA, penyerang memiliki satu buah nilai awal $h_0 \in \{0,1\}^n$ dan pesan *suffix* y. Tujuan dari serangan variasi CPA, penyerang dapat menemukan *prefix* p dan *suffix* x dengan $x \neq y$ sedemikian sehingga $f^+(h_0, p \parallel x) = f^+(h_0, p \parallel y)$. Pesan yang akan dicari *second preimage*-nya dengan variasi CIVA maupun CPA paling sedikit terdiri dari $d + 1$ blok pesan. Hal ini berarti bahwa pada blok pesan ke-$d + 1$, *second preimage* dari pesan telah ditemukan.

Kortelainen dan Kortelainen [10] menyatakan bahwa kompleksitas total dari serangan CIVA dan CPA masing-masing adalah $O(2^{\frac{n+d}{2}} + \sqrt{d} \cdot 2^{n-d})$. Jika $d = \frac{n}{3}$ maka kompleksitas dapat diminimalkan menjadi $O(\sqrt{n} \cdot 2^{\frac{2n}{3}})$. Apabila $\frac{n}{3} \notin Z$, maka dilakukan fungsi *ceiling* terhadap $\frac{n}{3}$ atau $\left\lceil \frac{n}{3} \right\rceil$.

2.5. Peluang dan Kompleksitas

Perhitungan peluang diperolehnya pasangan kolisi (h, x) dan (h', x') dilihat dari setiap proses *Step*. Pencarian pasangan kolisi dilakukan dengan membandingkan *hashing* pada proses *Step* dengan himpunan H_i, keluaran dari proses *Initialization*. Dengan demikian, pada proses *Step* terdapat pembandingan nilai *hash* sebanyak $|H_{j,k} \times f(A_{j,k}, M'_{j,k})| \geq 2^n$.

Perhitungan peluang diperolehnya pasangan kolisi dilakukan dengan menggunakan Persamaan 5 yang menjelaskan bahwa untuk semua bilangan real x berlaku persamaan berikut.

$$\lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n = e^x \dots\dots\dots(5).$$

Ketika dimasukkan nilai $x = -1$ pada Persamaan 5, nilai yang diperoleh yaitu $\left|\left(1 - \frac{1}{n}\right)^n\right| < e^{-1}$. Dengan demikian, peluang diperolehnya pasangan kolisi (PK) pada satu proses *Step* adalah

$$\begin{aligned} P(PK) &= 1 - \left(1 - \frac{1}{n}\right)^n \\ &= 1 - e^{-1} \\ &= \frac{e-1}{e} \dots\dots\dots(6). \end{aligned}$$

Kompleksitas yang akan dijelaskan merupakan kompleksitas waktu pada konstruksi struktur intan dan *second preimage attack*. Kompleksitas waktu konstruksi struktur intan diperoleh dari banyaknya proses *hashing* yang dilakukan pada proses *Initialization* $I(i)$, *Jump* (J_i), *Phase* ($P_{i,j}$), dan *Step*

$(S_{i,j,k})$. Kompleksitas total konstruksi struktur intan dinyatakan dalam $O\left(2^{\frac{n+d}{2}}\right)$.

Peluang ditemukannya *second preimage* diperoleh dari peluang pencarian pasangan kolisi pada proses konstruksi struktur intan dan pencarian blok pesan x' pada proses *second preimage attack*. Dengan demikian, peluang ditemukannya *second preimage* (SC) dari pesan y adalah $P(SC) = 1 - \left(\frac{1}{\frac{e}{e-1}.2^{n-d}}\right)^{n-d}$.

Kompleksitas *second preimage attack* diperoleh dari kompleksitas konstruksi struktur intan dan kompleksitas pencarian blok pesan x' sehingga memenuhi persamaan $f(h', x') = f^+(h_i, y_1 \| y_2 \| \cdots \| y_{12})$ membutuhkan 2^{n-d} pesan. Oleh karena itu, kompleksitas pencarian blok pesan x' adalah $O(\sqrt{d} \cdot 2^{n-d})$. Dengan demikian, kompleksitas total *second preimage attack* metode Kortelainen yaitu $O\left(2^{\frac{n+d}{2}} + \sqrt{d} \cdot 2^{n-d}\right)$.

3. METODOLOGI PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode kepustakaan dan eksperimen. Metode kepustakaan dilakukan dengan studi literatur pada referensi berupa buku, jurnal ilmiah, tesis, dan sumber lainnya yang berkaitan dengan teori serta konsep yang dibutuhkan. Teori dan konsep tersebut meliputi skema Davies-Meyer, algoritme SIMECK32/64, struktur intan, *second preimage attack*, serta peluang dan kompleksitas serangan. Metode eksperimen dilakukan dengan menerapkan algoritme SIMECK32/64 pada skema Davies-Meyer. Tahapan serangan yang dilakukan dijelaskan sebagai berikut.

a. Menentukan pesan $y = y_1 \| y_2 \| \cdots \| y_k$ sebagai pesan yang akan dicari *second preimage*-nya pada skema Davies-Meyer.
b. Mengonstruksi struktur intan metode Kortelainen.
    1) Membangkitkan himpunan H_d secara acak dengan $|H_d| = 2^d$.
    2) Melakukan proses *Jump* $(J_i, J_{i-1}, \dots, J_2)$ dimulai dari $i = d$ hingga $i = 2$.
    3) Melakukan proses *Jump* J_1 dengan membangkitkan himpunan pesan M_1 yang terdiri dari 2^{16} blok pesan untuk mendapatkan $x_1, x_2 \in M_1$ sedemikian sehingga $f(h_1, x_1) = f(h_2, x_2)$.
    4) Membentuk himpunan $B_1 = \{(h_1, x_1), (h_2, x_2)\}$; $H_0 = \{h_0\}$ dengan $h_0 = f(h_1, x_1) = f(h_2, x_2)$.
c. Membentuk struktur intan dengan himpunan $B_d, B_{d-1}, \dots, B_1$ yang merupakan keluaran setiap proses *Jump*. B_d merupakan daun dari struktur intan dan B_1 merupakan akar dari struktur intan atau nilai *hash* tunggal (h') yang dicari.
d. Melakukan *second preimage attack* variasi CIVA dan CPA dengan menggunakan hasil dari struktur intan.
    1) Mencari blok pesan x' dengan menggunakan nilai *hash* tunggal h'.
    2) Menghitung $f^+(h_i, y_1 \| y_2 \| \cdots \| y_{d+1})$ atau $f^+(h_0, p_i \| y_1 \| y_2 \| \cdots \| y_{d+1})$ dengan $1 \leq i \leq 2^d$ dan $h_i \in H_{11}$.
    3) Mencari h_i sedemikian sehingga $f(h', x') = f^+(h_i, y_1 \| y_2 \| \cdots \| y_{d+1})$ atau mencari *prefix* p_i sedemikian sehingga $f(h', x') = f^+(h_0, p_i \| y_1 \| y_2 \| \cdots \| y_{d+1})$.
    4) Mencari pesan z_1
e. Menghitung peluang dan kompleksitas dalam mengonstruksi struktur intan dan melakukan *second preimage attack* metode Kortelainen.
f. Menarik kesimpulan dari hasil penerapan struktur intan pada *second preimage attack* dan hasil dari *second preimage attack*.

4. PEMBAHASAN

4.1. Konstruksi Struktur Intan

Tahapan konstruksi struktur intan untuk variasi CIVA dan CPA dilakukan dengan tahapan yang sama tetapi dengan masukan yang berbeda. Struktur intan untuk serangan variasi CIVA dibangun dengan himpunan H_d yang dibangkitkan secara acak. Sementara itu, himpunan H_d yang digunakan untuk membangun struktur intan variasi CPA diperoleh dari hasil *hashing prefix* yang dibangkitkan.

Struktur intan dikonstruksi melalui d proses *Jump*, setiap *Jump* terdiri dari *i Phase* dengan $d \geq i \geq 1$, dan setiap *Phase* terdiri dari 2^{j-2} *Step* dengan $i \geq j \geq 2$. Setiap proses *Jump* ke-i membutuhkan himpunan masukan berupa H_d $(A_{i,0})$, pesan awal $(M_{i,0})$, dan pesan $M'_{j,k}$. Nilai i merupakan indeks proses *Jump*, nilai j merupakan indeks proses *Phase* dan k merupakan indeks proses *Step*. Banyaknya pesan awal $M_{i,0}$ yang dibangkitkan sebanyak $2^{\frac{32-i}{2}-1}$ dan banyaknya pesan $M'_{j,k}$ yang dibangkitkan sesuai dengan Persamaan 3. Nilai d yang dipilih yaitu $d = 11$ dan $d = 8$ untuk mengetahui perbedaan kompleksitas yang dijelaskan oleh Kortelainen serta keterbatasan perangkat pemrosesan.

Pada serangan variasi CPA, nilai H_d diperoleh dari *hashing prefix* dengan h_0 tetap. Langkah pertama yang dilakukan adalah membangkitkan dua blok *prefix* p_i sebanyak 2^d. Nilai d yang digunakan adalah $d = 8$. Langkah selanjutnya adalah melakukan *hashing* h_0 dengan 256 *prefix* $(p_1, p_2, \dots, p_{256})$ yang menghasilkan keluaran $f^+(h_0, p_1), f^+(h_0, p_2), \dots, f^+(h_0, p_{256})$. H_d yang digunakan sebagai masukan dalam membangun struktur intan ditunjukkan pada Tabel 1 dan Tabel 2.

Pembangkitan pesan $M'_{j,k}$ dilakukan di awal konstruksi struktur intan. Pra proses tersebut dilakukan untuk memastikan pesan tidak ada yang

bernilai sama dan efisiensi proses konstruksi struktur intan. Jumlah pesan $M'_{j,k}$ yang dibangkitkan merupakan akumulasi pesan yang dibutuhkan pada proses *Jump* J_i. Banyaknya pesan $|M_{i,0}|$, $|M'_{j,k}|$, dan $M'_{1,0}$ yang dibangitkan pada proses *Jump* $(J_{11}, J_{10}, \dots, J_2)$ ditunjukkan pada Tabel 3.

Tabel 1 Himpunan H_d sebagai pembangun struktur intan variasi CIVA

Variasi CIVA			
$d = 11$		$d = 8$	
No	H_{11}	No	H_8
1	$622c9c1f_{16}$	1	$fa46f7a6_{16}$
2	$2e354a8b_{16}$	2	$4080169e_{16}$
⋮	⋮	⋮	⋮
512	$0b9100b5_{16}$	128	$6224f0b3_{16}$
⋮	⋮	⋮	⋮
1023	$22897d4b_{16}$	255	$d85dfefa_{16}$
1024	$df53a066_{16}$	256	$051471cb_{16}$

Tabel 2 Himpunan H_d sebagai pembangun struktur intan variasi CPA

Variasi CPA	
$d = 8$	
No	H_8
1	$aafe4513_{16}$
2	$1bb61dbd_{16}$
⋮	⋮
128	$54b8e213_{16}$
⋮	
255	$374e20bd_{16}$
256	$3ce0ab16_{16}$

Tabel 3 Jumlah masukan paada proses *Jump* J_i

i	$\lvert M_{i,0}\rvert$	$\sum_{j=i}^{2}\sum_{k=0}^{2^{j-2}-1}\lvert M'_{j,k}\rvert$	$\lvert M'_{1,0}\rvert$	Total Pesan
11	724	65241	65536	131502
10	1024	64005	65536	130565
9	1448	62452	65536	129436
8	2048	60387	65536	127971
7	2896	57529	65536	125961
6	4096	53545	65536	123177
5	5792	47983	65536	119312
4	8192	40298	65536	114026
3	11585	29901	65536	107022
2	16384	16384	65536	98304

Berikut ini akan dijelaskan ilustrasi proses $P_{11,11}$ (Phase) dan $S_{11,11,0}$ (*Step*). Masukan pada proses $P_{11,11}$ adalah himpunan $A_{11,0}$ dan $M_{11,0}$. Selanjutnya, masukan ini diproses pada $S_{11,11,0}$. Proses $S_{11,11,0}$ hanya memperoleh satu pasangan kolisi (h, x) dan (h', x') yang ditunjukkan pada Tabel 4.

Tabel 4 Pasangan kolisi pada $S_{11,11,0}$

No.	Nilai *IV*	Pesan	Nilai *Hash*
1	$622c9c1f_{16}$	$0df0fcd4d1182061_{16}$	$5bf2674e_{16}$
	$88928e6a_{16}$	$ec43d5b791cd6901_{16}$	

Proses *Step* selanjutnya dilakukan dengan tahapan yang sama. Proses *Phase* dan *Step* dilakukan hingga $P_{11,1}$ dan $S_{11,1,0}$ atau proses J_1. Hasil dari proses J_1 yang selanjutnya disebut dengan nilai *hash* tunggal ditunjukkan pada Tabel 5.

Pencarian pasangan kolisi pada struktur intan variasi CIVA dan CPA dengan $d = 8$ dilakukan dengan cara serupa. Masing-masing struktur intan variasi CIVA dan CPA membutuhkan 256 masukan (H_8) yang telah ditunjukkan pada Tabel 1 dan Tabel 2. Nilai *hash* tunggal pada struktur intan variasi CIVA dan CPA dengan $d = 8$ diperoleh setelah melakukan 8 proses *Jump* $(J_8, J_7, \dots, J_1)$. Nilai *hash* tunggal yang diperoleh dari struktur intan variasi CIVA dan CPA dengan $d = 8$ ditunjukkan pada Tabel 6.

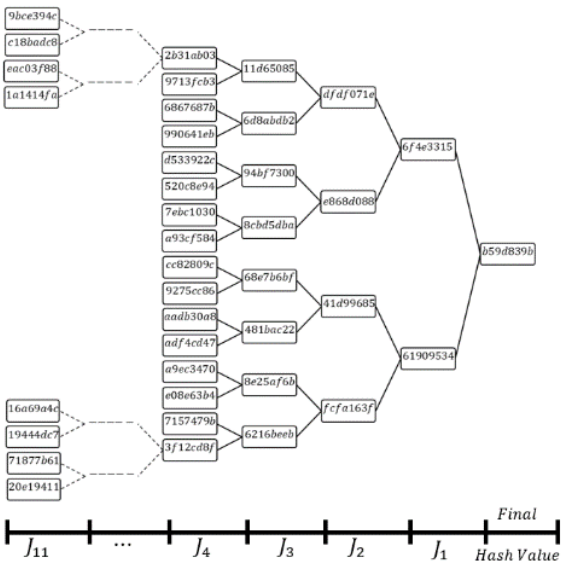
Tabel 5 Keluaran proses J_1

$A_{1,0}$	Pesan M'_1	Nilai *Hash* Tunggal
$6f4e3315_{16}$	$cb7577b4988d255f_{16}$	$b59d839b_{16}$
61909534_{16}	$467080d918945a64_{16}$	

Tabel 6 Nilai *hash* tunggal struktur intan variasi CIVA dan CPA dengan $d = 8$

	$A_{1,0}$	Pesan M'_1	Nilai *Hash* Tunggal
CIVA	$283b5eae_{16}$	$472e8f459aba3fa3_{16}$	33457770_{16}
	$227b9646_{16}$	$24d61d71427aedbb_{16}$	
CPA	$61fd4287_{16}$	$85c992a607023f84_{16}$	$4e4854f4_{16}$
	$ef75d937_{16}$	$81af272808f7269b_{16}$	

Struktur intan dibentuk dengan menghubungkan variabel berantai yang dihasilkan pada proses *Jump* $(J_i, J_{i-1}, \dots, J_1)$. Ilustrasi dari skema struktur intan yang ditunjukkan adalah struktur intan variasi CIVA dengan $d = 11$. Struktur intan tersebut ditunjukkan padaGambar 3.



Gambar 3 Struktur intan variasi CIVA dengan $d = 11$

4.2. *Second Preimage Attack* Variasi CIVA

Serangan ini disebut dengan CIVA karena untuk mencari *second preimage* dari blok pesan y yang diketahui, memilih nilai h_i dalam himpunan H_d dengan $d = 11$ dan $d = 8$ sebagai nilai awal hingga ditemukan *second preimage* dari y. Pada serangan ini, nilai yang diketahui adalah blok pesan y, sedangkan nilai awal (h_0) tidak diketahui. Pesan y yang akan dicari nilai *second preimage*-nya ditunjukkan pada Tabel 7.

Ilustrasi serangan yang dijelaskan adalah *second preimage attack* variasi CIVA dengan $d = 11$. Serangan diawali dengan membangkitkan pesan x'

sebanyak 2^{21} untuk mencari blok pesan x' yang memenuhi persamaan $f(h', x') = f^+(h_i, y_1 \parallel y_2 \parallel \cdots \parallel y_{12})$ dengan h_i merupakan nilai awal dalam himpunan H_{11}. Serangan ini memperoleh dua nilai x' dan h_i berbeda. Pesan x sebagai *second preimage* dari y ditunjukkan pada Tabel 8.

Tabel 7 Sampel 13 blok pesan

Blok ke-	Pesan
1	$04bcae89f43203b1_{16}$
2	$5bb42684ff19e0e0_{16}$
3	$91a3bdc26d7344c7_{16}$
4	$d94b1a8aeb993902_{16}$
5	$f6f3646f18a312ce_{16}$
6	$5e62216031ed5c76_{16}$
7	$6be34f54836b2663_{16}$
8	$b17de10d6b799276_{16}$
9	$54061324455de0ab_{16}$
10	$cadb064e5eac588b_{16}$
11	$fc88fdd070b5da6a_{16}$
12	$0dec2c6f02b60cdb_{16}$
13	$88cefec802c0d4f1_{16}$

Tabel 8 Pesan x sebagai *second preimage* dari y

Nilai h_i		Blok Pesan x ke-1	Nilai *Hash*
$c098c19c_{16}$	z_1	$7a57a5da7d2d0cb4_{16}$	$0d38579e_{16}$
		$a314fc89d7b0699a_{16}$	
		$4d462b870543c6c2_{16}$	
		$3800a5eb3c834894_{16}$	
		$65f69e3d1438d811_{16}$	
		$869a0d99c009a98c_{16}$	
		$d11aca96cf5c3e7f_{16}$	
		$cf0cef07c6924d99_{16}$	
		$771d00a6f3a39a02_{16}$	
		$5105dc2e17366c9b_{16}$	
		$467080d918945a64_{16}$	
	x'	$d79dffd7cb4accb4_{16}$	
	y_{13}	$88cefec802c0d4f1_{16}$	
$b02e6f15_{16}$	z_1	$2c286f70f3269653_{16}$	$f236c825_{16}$
		$bd74a19568241522_{16}$	
		$779b8f1ebb7f17a4_{16}$	
		$f01b5c28c124c7a4_{16}$	
		$5cce5b652698e666_{16}$	
		$07a881274e248b27_{16}$	
		$2275bd906de7c304_{16}$	
		$b7c352d5a696ad74_{16}$	
		$c0418e2d00fd4246_{16}$	
		$5105dc2e17366c9b_{16}$	
		$467080d918945a64_{16}$	
	x'	$a1074a248209c964_{16}$	
	y_{13}	$88cefec802c0d4f1_{16}$	

Tabel 9 Pesan x sebagai *second preimage* y (CIVA $d = 8$)

Nilai h_i		Blok Pesan x	Nilai *Hash*
$3a4797ef_{16}$	z_1	$828d514854953ea2_{16}$	$2d840f17_{16}$
		$e2e4e50510acbf2c_{16}$	
		$ffa41778737a953d_{16}$	
		$4087c2810a563ca4_{16}$	
		$69781ab55d1e9e9b_{16}$	
		$29fd0cf6cafac07c_{16}$	
		$eb5fb10b5bca09ed_{16}$	
		$24d61d71427aedbb_{16}$	
	x'	$b372494b1ab81a44_{16}$	
	y_{10}	$5105dc2e17366c9b_{16}$	
	y_{11}	$467080d918945a64_{16}$	
	y_{12}	$d79dffd7cb4accb4_{16}$	
	y_{13}	$88cefec802c0d4f1_{16}$	

4.3. Second Preimage Attack Variasi CPA

Second preimage attack variasi CPA dilakukan dengan memanfaatkan satu nilai awal h_0 dan pesan *suffix* y. *Second preimage* yang dicari pada serangan variasi CPA merupakan pesan berupa *suffix* x dengan *prefix* p. *Suffix* x terdiri dari 13 blok pesan sesuai dengan syarat yang dinyatakan Kortelainen dan Kortelainen yaitu $k > d + 1$. Tahap awal *second preimage attack* variasi CPA sama dengan tahap awal *second preimage attack* variasi CIVA. Akan tetapi, jumlah pesan x' yang dibutuhkan hingga ditemukan blok pesan x' dan p_i yang memenuhi persamaan $f(h', x') = f^+(h_0, p_i \parallel y_1 \parallel y_2 \parallel \cdots \parallel y_9)$ berbeda. Serangan ini membutuhkan sampel x' sebanyak $\sqrt{8} \cdot 2^{21}$ hingga diperoleh pesan x' yang dicari. Proses ini mendapatkan satu pesan x' dan satu pesan *prefix* p_i. Blok pesan x sebagai *second preimage* dari pesan y diperoleh dengan menggabungkan *prefix* p_i, blok pesan x', blok pesan z_1, dan blok pesan $y_{10} \parallel y_{11} \parallel y_{12} \parallel y_{13}$. *Prefix* p_i merupakan dua blok *prefix* yang menghasilkan nilai *hash* sebagai daun pada struktur intan variasi CPA dengan $d = 8$. Blok pesan x sebagai *second preimage* y ditunjukkan pada Tabel 10.

Tabel 10 Pesan x sebagai *second preimage* y (*CPA* $d = 8$)

Nilai h_0		Blok Pesan x	Nilai *Hash*
$f8b9de11_{16}$	*Prefix* p_i	$615e107aa5e65e96_{16}$	$44b6fec0$
		$48c688753cb1ee56_{16}$	
	z_1	$f0104dc8199f7be7_{16}$	
		$9193b7b8410e61f0_{16}$	
		$8f8d250c281294e4_{16}$	
		$13d43b4f86439d24_{16}$	
		$dcef8bb9f64e9b0c_{16}$	
		$64ed161016590012_{16}$	
		$dea82a4f4bd96209_{16}$	
		$81af272808f7269b_{16}$	
	x'	$7169d3ad4523d8bc_{16}$	
	y_{10}	$5105dc2e17366c9b_{16}$	
	y_{11}	$467080d918945a64_{16}$	
	y_{12}	$d79dffd7cb4accb4_{16}$	
	y_{13}	$88cefec802c0d4f1_{16}$	

4.4. Peluang dan Kompleksitas

Perhitungan peluang dan kompleksitas konstruksi stuktur intan dilakukan sesuai dengan tahapan yang dijelaskan pada poin 2.5. Ilustrasi yang akan diberikan adalah peluang ditemukannya pasangan kolisi pada $S_{11,11,0}$ variasi CIVA dengan $d = 11$. $H_{11,0}$ merupakan himpunan nilai *hash* dari $f(A_{11,0}, M'_{11,0})$ sehingga banyaknya anggota dalam $H_{11,0}$ adalah $|H_{11,0}| = |A_{11,0}||M_{11,0}|$. Maka diperoleh $|H_{11,0}| = 1482752$. Sementara banyaknya anggota himpunan pembanding yaitu $|f(A_{11,0}, M'_{11,0})| = 4096$. Penelitian ini menggunakan algoritme SIMECK32/64 sehingga peluang kolisi setiap nilai keluaran adalah $\frac{1}{2^{32}}$ karena terdapat 2^{32} kemungkinan nilai yang dihasilkan. Dengan demikian, peluang diperolehnya pasangan kolisi (PK) pada $S_{11,11,0}$ dihitung sesuai dengan Persamaan 6, yaitu

$$P(PK) = 1 - \left(1 - \frac{1}{2^{32}}\right)^{|H_{j,k}|\left|f\left(A_{j,k}, M'_{j,k}\right)\right|}$$
$$= 1 - 0{,}2432$$
$$= 0{,}7568$$

Perhitungan peluang untuk proses *Step* selanjutnya dilakukan dengan cara yang sama. Nilai yang membedakan setiap perhitungan peluang proses *Step* adalah banyaknya $|H_{j,k}|$ dan $|f(A_{j,k}, M'_{j,k})|$ yang dibutuhkan. Nilai $|H_{j,k}|, |f(A_{j,k}, M'_{j,k})|$, dan peluang diperolehnya pasangan kolisi pada proses *Step* ditunjukkan pada Tabel 11.

Tabel 11 Peluang diperolehnya pasangan kolisi pada *Jump* J_{11}

Phase	*Step*	$\|H_{j,k}\|$	$\|f(A_{j,k}, M'_{j,k})\|$	Peluang $(P(PK))$
$P_{11,11}$	$S_{11,11,0}$	1482752	4096	0,7568
	$S_{11,11,1}$	1483350	4092	0,7566
⋮	⋮	⋮	⋮	⋮
$P_{11,8}$	$S_{11,8,0}$	524288	8192	0,6321
	$S_{11,8,1}$	524510	8382	0,6407
⋮	⋮	⋮	⋮	⋮
$P_{11,5}$	$S_{11,5,0}$	185376	23200	0,6326
	$S_{11,5,1}$	185370	23190	0,6324
⋮	⋮	⋮	⋮	⋮
$P_{11,3}$	$S_{3,3,0}$	92688	61792	0,7364
$P_{11,2}$	$S_{2,2,0}$	65536	65536	0,6321

Tabel 11 menunjukkan bahwa setiap *Step* memiliki peluang ditemukannya pasangan kolisi yang mendekati nilai $\frac{e-1}{e}$ atau 0,632 yang telah dijelaskan pada poin 2.5. Dengan demikian, peluang pencarian pasangan kolisi yaitu $P(PK) \approx 0{,}632$.

Perhitungan kompleksitas yang dilakukan pada proses *Jump* sesuai dengan Persamaan 5 Kompleksitas setiap proses *Jump* dan kompleksitas total untuk ketiga struktur intan ditunjukkan pada Tabel 12.

Tabel 12 Kompleksitas konstruksi struktur intan

Kompleksitas		$I(11), \ldots, I(2)$	$J_i, J_{i-1}, \ldots, J_2$	J_1	Total
Struktur Intan Variasi CIVA	$d = 11$	$2^{22,5}$ komputasi	$3 \cdot 2^{23}$ komputasi	2^{17} komputasi	2^{25} komputasi
	$d = 8$	$3 \cdot 2^{19}$ komputasi	$2^{23,5}$ komputasi	2^{16} komputasi	$3 \cdot 2^{22,5}$ komputasi
Struktur Intan Variasi CPA	$d = 8$	$3 \cdot 2^{19}$ komputasi	2^{23} komputasi	2^{16} komputasi	$2^{23,5}$ komputasi

Peluang diperolehnya *second preimage* pada *second preimage attack* variasi CIVA dengan $d = 11$ yaitu

$$P(CIVA11) = 1 - \left(\frac{1}{2\frac{e}{e-1}2^{21}}\right)^{2^{21}}.$$

Sementara itu, variasi CIVA dan variasi CPA dengan $d = 8$ masing-masing memperoleh satu blok pesan x' dari 2^{21} dan $\sqrt{8} \cdot 2^{21}$ sampel pesan. Peluang diperolehnya *second preimage* dari kedua serangan ini yaitu

$$P(CIVA8) = 1 - \left(\frac{1}{\frac{e}{e-1}2^{21}}\right)^{2^{21}}; \text{ dan}$$
$$P(CPA8) = 1 - \left(\frac{1}{\frac{e}{e-1}\sqrt{8}\cdot 2^{21}}\right)^{\sqrt{8}\cdot 2^{21}}.$$

Kompleksitas *second preimage attack* metode Kortelainen diperoleh dari penjumlahan kompleksitas konstruksi struktur intan dan pencarian blok pesan x'. Serangan variasi CIVA dengan $d = 11$ membutuhkan sampel pesan sebanyak 2^{21}. Sementara untuk serangan variasi CIVA dan CPA dengan $d = 8$ masing-masing membutuhkan 2^{21} dan $\sqrt{8} \cdot 2^{21}$. Dengan demikian, kompleksitas *second preimage attack* metode Kortelainen variasi CIVA dengan nilai $d = 11$ adalah $2^{25} + 2^{20}$ komputasi dan dengan nilai $d = 8$ adalah $3 \cdot 2^{22,5} + 2^{21}$ komputasi. Kompleksitas *second preimage attack* variasi CPA dengan $d = 8$ adalah $2^{23,5} + \sqrt{8} \cdot 2^{21}$ komputasi. Apabila dibandingkan dengan *second preimage resistance*, *second preimage* metode Kortelainen memiliki kompleksitas yang lebih kecil dari *brute force attack* yaitu $O(2^n)$. Dengan demikian, serangan metode ini lebih efektif daripada *brute force attack.*

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan dari penelitian ini adalah:

a. Kompleksitas yang dibutuhkan untuk konstruksi struktur intan variasi CIVA dengan $d = 11$ dan $d = 8$ serta variasi CPA dengan $d = 8$ yaitu 2^{25} komputasi, $3 \cdot 2^{22,5}$ komputasi, dan $2^{23,5}$ komputasi.
b. Penerapan dua variasi *second preimage attack* metode Kortelainen pada skema Davies-Meyer berbasis SIMECK32/64 memperoleh *second preimage* dari blok pesan y. Variasi CIVA dengan $d = 11$ memperoleh dua *second preimage* sedangkan variasi CIVA dan CPA dengan $d = 8$ masing-masing memperoleh satu pesan x sebagai *second preimage* dari y.
c. Kompleksitas *second preimage attack* metode Kortelainen yang diperoleh memiliki kompleksitas yang lebih baik daripada *brute force attack*.

5.2. Saran

Saran untuk penelitian lebih lanjut yang dapat dilakukan:

a. Melakukan *second preimage attack* metode Kortelainen pada skema fungsi *hash* lain yang berbasis Merkle-Damgard seperti Miyaguchi-Preneel dan Matyas-Meyer-Oseas.
b. Memanfaatkan struktur intan untuk serangan fungsi *hash* lainnya.

REFERENSI

[1] C. Paar dan J. Pelzl, *Understanding Cryptography: A Textbook for Student and Practitioners*. Springer Science & Business Media, 2009.

[2] R. C. Merkle, "A Fast Software One-Way Hash Functions," *Journla Cryptol.*, hal. 43–58, 1989.

[3] I. B. Damgard, "A Design Principle for Hash Functions," in *CRYPTO '89*, 1989, hal. 416–427.

[4] A. J. Menezes, P. C. Van Oorschot, dan S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[5] B. Preneel, R. Govaerts, dan J. Vandewalle, "Hash Functions based on Block Ciphers : A Synthetic Approach," *Adv. Cryptology-CRYPTO '93, LNCS*, hal. 368–378, 1994.

[6] J. Black, P. Rogaway, dan T. Shrimpton, "Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV," hal. 320–335, 2002.

[7] E. Andreeva *et al.*, "Second Preimage Attacks on Dithered Hash Functions," no. Berlin, Heidelberg: Springer, hal. 270–288, 2008.

[8] J. Kelsey dan T. Kohno, "Herding Hash Functions and the Nostradamus Attack," *EUROCRYPT*, hal. 183–200, 2006.

[9] S. R. Blackburn dan D. R. Stinson, "On The Complexity of The Herding Attack and Some Related Attacks on Hash Functions," hal. 171–193, 2012.

[10] T. Kortelainen dan J. Kortelainen, "On Diamond Structures and Trojan Message Attacks," 2013, hal. 524–539.

[11] T. Kortelainen dan J. Kortelainen, "New Second Preimage Attack Variants against the MD-Structure," 2014, hal. 98–110.

[12] Zhu, Bo dan G. G. Yang, G., B. Zhu, V. Suder, M. D. Aagaard, "The Simeck Family of Lightweight Block Ciphers," 2015, hal. 307–329.