FIXED POINT ATTACK PADA *SIMPLIFIED* IDEA DENGAN SKEMA DAVIES-MEYER

Agus Winarno[1)], **Syamsi Nurdiansah**[2)], **Sigit Setiono**[3)]

[1,2,3)] *Lembaga Sandi Negara*
Jl. Harsono RM 70 Ragunan, Pasar Minggu, Jakarta Selatan 12550
Email : agus.winarno@bssn.go.id[1)], syamsi.nurdiansah@bssn.go.id [2)], sigit.setiono@bssn.go.id [3)]

Abstrak

fixed point attack adalah serangan yang dapat diterapkan pada skema Davies-Meyer, yaitu skema fungsi hash yang didesain menggunakan konsep yang sama dengan struktur Merkle-Damgard dengan penambahan operasi feed-forward dari input ke output.

Pada paper ini dilakukan fixed point attack pada simplified IDEA pada skema Davies-Meyer. Hasil message expansion attack menunjukkan bahwa simplified IDEA tidak memiliki sifat collision resistance karena pada fixed point attack penyerang dapat menemukan kolisi dengan probabilitas $4{,}5\times10^{-5}$

Kata kunci: *fixed point attack, simplified IDEA, Davies-Meyer.*

1. Pendahuluan

Pada tahun 1989, Ralph C.Merkle memperkenalkan desain fungsi *hash* yang bertujuan untuk memproses pesan yang panjangnya sembarang ke pesan yang panjangnya tetap. Desain ini memiliki konsep yang sama dengan yang diperkenalkan Ivan B. Damgard sehingga struktur ini lebih dikenal dengan nama Merkle-Damgard [1][2]. Struktur Merkle-Damgard yang menggunakan *block cipher* sebagai fungsi internalnya memiliki kerentanan terhadap *meet-in-the-middle attack*. Solusi dari permasalahan tersebut maka diciptakan skema *hash* berbasis *block cipher* yang didesain berdasarkan struktur Merkle-Damgard dengan menambahkan operasi *feed-forward* dari *input* ke *output*-nya. Skema tersebut terdiri dari Davies-Meyer, Matyas-Meyer-Oseas, Miyaguchi-Preneel [3]. Dari ketiga skema tersebut hanya Davies-Meyer yang dapat diterapkan pada *block cipher* yang memiliki panjang blok pesan dan kunci yang berbeda [4]. Salah satu algoritma yang memiliki panjang kunci yang berbeda dengan panjang pesan adalah IDEA [5].
Pada perkembangan selanjutnya, struktur Davies-Meyer masih mewarisi kelemahan struktur Merkle-Damgard, yaitu rentan terhadap *fixed point attack* [6][7], sehingga pada paper ini akan dilakukan penelitian penerapan *fixed point attack*.
Fixed point attack pada skema Davies-Meyer dengan *block cipher* IDEA membutuhkan komputasi yang cukup besar, sehingga dipilih *Simplified* IDEA sebagai objek penelitiannya. *Simplified* IDEA merupakan versi mini dari algoritma *block cipher* IDEA yang menggunakan ukuran blok *input* sepanjang 16 bit, ukuran kunci 32 bit, dan jumlah *round* yang digunakan sebanyak 5 *round* [8]. Selanjutnya, hasil kolisi akan dianalisi berdasarkan sifat *collision resistance*.

2. Landasan Teori

Kriptografi[9]

Kriptografi adalah salah teknik matematika yang berkaitan dengan aspek keamanan informasi yang terdiri dari kerahasian, keutuhan data, otentikasi entitas, dan otentikasi data. Teknik kriptografi Menyediakan layanan Kerahasiaan (*confidentiality*) yaitu layanan keamanan informasi yang bertujuan untuk menjaga informasi supaya tidak diketahui pihak lain yang tidak memiliki wewenang. Keutuhan data (*Data Integrity*) bertujuan memberikan layanan agar data yang dimiliki atau ditransmisikan tidak mengalami manipulasi seperti penambahan, pengurangan, dan penggantian pada isi informasinya, sedangkan otentikasi adalah layanan yang bertujuan untuk mengidentifikasi entitas dan data yang ada masih otentik sesuai dengan data awal.
Teknik kriptografi yang dikelompokkan dapat menjadi 3 bagian besar yaitu fungsi *hash* (*hash function*), Algoritma kunci Simetris (*symmetric key*), dan Algoritma kunci asimetris (*asymmetric key*). Fungsi *hash* adalah algoritma kriptografi yang bertujuan untuk memetakan input yang panjangnya sembarang pada *output* yang panjangnya tetap. Sementara itu, algoritma kunci simetris adalah algoritma kriptografi yang dalam proses enkripsi dan dekripsi menggunakan kunci rahasia yang sama. Algoritma kunci simetris adalah algoritma kriptografi yang dalam proses enkripsi dan dekripsi menggunakan dua kunci yang berbeda yaitu kunci publik dan kunci privat. Algoritma kunci asimetris memiliki kunci publik yang dapat dipublikasikan secara umum dan dapat diketahui semua orang dan kunci privat yang bersifat rahsia dan hanya diketahui oleh pemiliknya.

Block Cipher

Block cipher merupakan suatu skema enkripsi yang membagi *plaintext* ke dalam blok-blok *plaintext* yang tetap dan membutuhkan kunci rahasia untuk mengubah *plaintext* tersebut menjadi *ciphertext*. Untuk mendapatkan kembali *plaintext* yang dienkripsi maka dibutuhkan kunci rahasia yang sama untuk melakukan proses dekripsi [10]. Oleh karena itu *block cipher* termasuk dalam salah satu algoritma sistem kunci simetris. Algoritma kunci simetris adalah algoritma

kriptografi yang dalam proses enkripsi dan dekripsi menggunakan kunci rahasia yang sama[9]

***Simplified* IDEA [8]**

Algoritma *block cipher Simplified* IDEA merupakan versi mini dari algoritma IDEA. *Simplified* IDEA memiliki panjang blok *input* dan *output* sepanjang 16 bit dan menggunakan kunci sepanjang 32 bit. Jumlah *round* yang digunakan pada algoritma *Simplified* IDEA adalah sebanyak 5 *round*. Operasi yang digunakan dalam algoritma terdiri dari XOR, penjumlahan modul 2^{16}, dan perkalian modul 2^{17}. Operasi-operasi tersebut digunakan untuk menyusun komponen fungsi *key mixing* dan *Multiplication Addition* (MA).

Proses enkripsi dan dekripsi pada *simplified* IDEA menggunakan struktur yang sama. Namun, pada proses dekripsi dibutuhkan inversi dari subkunci yang digunakan dalam proses enkripsi. Inversi pada operasi perkalian dan penjumlahan dapat diperoleh dengan menggunakan *lookup table* penjumlahan modulo 16 dan *lookup table* perkalian modulo 17. Contoh perhitungan enkripsi dan dekripsi dapat dilihat pada [8]. Enkripsi pesan $9CAC_H$ dengan kunci $DC6F3F59_H$menghasilkan *ciphertext* $BB4B_H$.

Fungsi Hash

Skema Fungsi *hash* adalah fungsi yang memetakan *plaintext* yang panjangnya tidak tetap kedalam sebuah *fingerprint* yang panjangnya tetap dari suatu *plaintext*. Secara umum fungsi *hash* bertujuan untuk menyediakan integritas suatu data, namun sekarang fungsi *hash* juga dapat digunakan pada skema komitmen, *key derivation*, *pseudorandom number generation*[10].

Fungsi *hash* juga dapat didefinisikan sebagai fungsi *h* yang mentransformasi *plaintext M* yang panjangnya berubah-ubah ke nilai *hash* yang panjang *m*. Syarat algoritma fungsi *h* yang digunakan harus efisien secara komputasi. Fungsi *hash* yang digunakan untuk penunjang pengamanan data adalah *cryptographic hash function*. *Cryptographic hash function* bertujuan untuk menyediakan jaminan integritas data dengan cara membentuk *fingerprint* atau *message digest* dari suatu data. Jika data tersebut ditempatkan di tempat yang tidak aman, integritasnya dapat dicek dari waktu ke waktu dengan merekonstruksi *fingerprint* dari data tersebut[11]. Sifat-sifat yang harus dimiliki suatu fungsi *hash*, antara lain[9] :

- Kompresi : fungsi *hash* h memetakan *input* M dengan panjang bit sembarang, menjadi *output* $h(M)$ dengan panjang n bit yang tetap.
- Mudah dalam komputasi : diberikan fungsi *hash* h dan *input* M, maka mudah untuk menghitung nilai *output* $h(M)$.

Fungsi *hash* memiliki dua kategori yaitu *keyed hash function* atau *Message Authentication Codes* (MAC's) yang membutuhkan kunci rahasia dan *plaintext* sebagai *input*nya dan *unkeyed hash function* atau *Modification Detection Codes* (MDC's) yang hanya membutuhkan *plaintext* sebagai *input*nya .

Skema Davies-Meyer [9]

Skema ini menggunakan *input*an *plaintext* sebagai kunci *block cipher*, sedangkan *output hash* dari iterasi sebelumnya digunakan sebagai *plaintext* dalam proses enkripsi selanjutnya oleh *block cipher* pembangunnya. Berikut adalah penjelasan algoritma Davies-Meyer :

Algoritma 1. **Skema Davies-Meyer [9]**

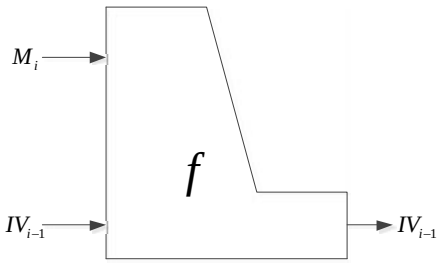
Input : *string* bit M.
Ouput : kode *hash* n-bit.

1. *Input* M dibagi menjadi n-bit blok dan menambahkan *padding* jika dibutuhkan untuk melengkapi blok terakhir. Notasikan *plaintext* yang telah di-*padding* yang terdiri dari t blok dengan masing-masing blok sebanyak n-bit : $M_1M_2 \ldots\ldots M_t$. Nilai *IV* sebanyak n-bit harus ditentukan sebelumnya.
2. *Output*nya adalah H_t yang didefinisikan sebagai :
   $H_0 = IV$; $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$, $1 \le i \le t$.

Skema Davies-Meyer dapat dilihat pada [9].

***Fixed Point Attack* [6]**

Fixed point dalam fungsi kompres $f(IV_{i-1},M_i) = IV_i$ akan terjadi jika pasangan (IV_{i-1},M_i) dapat memenuhi keadaan $IV_{i-1} = f(IV_{i-1},M_i)$, dengan *IV* adalah *initial vector* dan m_i adalah *plaintext* ke-i dalam suatu fungsi kompresi. Keadaan tersebut dapat diilustrasikan pada Gambar 1.



Gambar 1. *Fixed Point pada fungsi kompresi* ***[6]***

Keberadaan *plaintext* M_i tersebut tidak akan memberikan pengaruh pada perhitungan *hashing plaintext* M, karena *plaintext* M_i yang disisipkan pada *plaintext* M akan menghasilkan IV_i yang sama dengan *IV* sebelumnya.

3. Pembahasan

Metode Penelitian

Sampel

Sampel yang digunakan dalam penelitian ini terdiri dari 16 *input* ekstrim dan 16 *input pseudorandom*. *Input ekstrim* merupakan representase dari sampel yang memiliki pola tetap, sedangkan *input pseudorandom* merupakan representasi dari sampel yang tidak memiliki pola tetap. *Input message expansion attack* yang digunakan adalah sepanjang 512 bit, *input* ekstrim terdiri dari $0000…0000_H$ sampai $FFFF…FFFF_H$, sedangkan *input pseudorandom* yang digunakan merupakan hasil dari *pseudorandom generator* dengan *seed* berdasarkan waktu dari Sistem Operasi Windows 7 Ultimate. *Initial Vector* yang digunakan dalam *message expansion attack* adalah 1234_H.

Input pesan yang digunakan pada *fixed point attack* adalah $1111…1111_H$ (512 bit), sedangkan *IV* yang digunakan terdiri dari 16 *IV* ekstrim dan 16 *IV pseudorandom* dengan modifikasi minor sebanyak 2^{16}, *IV* ekstrim terdiri dari 0000_H sampai $FFFF_H$, sedangkan *IV pseudorandom* yang digunakan merupakan hasil *pseudorandom generator* dengan *seed* yang berasal dari waktu.

Fixed Point Attack

Pengujian tingkat keamanan selanjutnya dilakukan dengan *fixed point attack.* Serangan ini digunakan untuk menemukan *input plaintext* yang menghasilkan nilai *hash* yang sama dengan IV sebelumnya. *Plaintext-plaintext* yang didapatkan dari *fixed point attack* dapat digunakan untuk mengkonstruksi *plaintext-plaintext* yang saling berkolisi dengan cara menempatkan *plaintext* tersebut blok terakhir pada *plaintext* yang akan dipalsukan. Langkah-langkah yang harus dilakukan untuk memperoleh data hasil *fixed point attack* adalah sebagai berikut:

- Penyerang memilih *initial vector* awal (IV_0) dan *plaintext* M sepanjang 512 bit, *plaintext* dibagi menjadi 16 blok *plaintext* $M = M_1, M_2, \ldots, M_{16}$.
- Melakukan *hashing plaintext* M dengan IV_0 sehingga diperoleh nilai *hash* $H = h(M)$. Fungsi *hash* h itu sendiri merupakan proses *chaining* dari fungsi kompresi $IV_i = f(IV_{i-1}, M_i)$ dengan $i = 1, 2, \ldots, 16$ sehingga diperoleh $H = IV_{16} = f(IV_{15}, M_{16})$.
- Penyerang menambahkan satu blok *plaintext* M_{17} dengan modifikasi minor sebanyak 2^{16}.
- Lakukan proses kompresi pada *plaintext* blok ke tujuh belas agar diperoleh $f(IV_{16}, M_{17}) = IV_{16}$. *Plaintext* M_{17} yang memenuhi syarat tersebut merupakan *pesan fixed point*.

Data hasil *fixed point attack* yang diperoleh disajikan dalam bentuk tabel yang menunjukkan adanya *plaintext-plaintext* yang memenuhi sifat *fixed point* sehingga jika *plaintext* tersebut diletakkan di *round* ke 17 akan menghasilkan nilai *hash* yang sama dengan nilai *hash* dari *round* ke 16. *Plaintext fixed point* tersebut dapat digunakan penyerang untuk membentuk *plaintext-plaintext* berkolisi dengan *plaintext* asli (*plaintext* pada 16 *round* pertama)..

Simulasi

Simulasi serangan *fixed point attack* ini menggunakan bahasa pemrograman c. Komputasi yang digunakan dengan menggunakan prosesor dual core AMD A4 dengan RAM 2GB. Selanjutnya untuk pengolahan hasil dan penyajian menggunakan menggunakan Ms. Excel.

Hasil Penelitian *Fixed Point Attack*

IV ekstrim

Hasil *fixed point* dari penerapan *simplified* IDEA pada skema Davies Meyer dengan input ekstrim dapat dilihat pada Tabel 1.

Tabel 1. *Hasil Fixed point attack Penerapan simplified IDEA pada Skema Davies Meyer dengan Initial vector Ekstrim*

IV (4 Heksa)	*Plaintext* (136 Heksa)	*Hash*	*Fixed Point* (8 Heksa)
0000	11111….1111	3917	000000F2
			0000C70A
			0000F4F2
1111	11111….1111	2EAE	000021EF
2222	11111….1111	3C7B	-
3333	11111….1111	A616	-
4444	11111….1111	BB6E	-
5555	11111….1111	6F49	0000C35D
6666	11111….1111	6D2F	-
7777	11111….1111	D71D	-
8888	11111….1111	28A7	00009A64
9999	11111….1111	D20F	-
AAAA	11111….1111	3D7F	-
BBBB	11111….1111	C3DF	00000D02
			0000B9F4

			0000BBB7
CCCC	11111....1111	9CC0	-
DDDD	11111....1111	B777	000045AE
EEEE	11111....1111	13F6	0000614D
FFFF	11111....1111	3A15	0000CB4D

Berdasarkan Tabel 1. dapat dilihat bahwa penyerang dapat menemukan *fixed point* dengan jumlah maksimal sebanyak 3 buah dari 65536 variasi *input* pada blok terakhir. Probabilitas yang dibutuhkan untuk memperoleh *fixed point* adalah 0,000045. *Fixed point* yang diperoleh dapat dipergunakan penyerang untuk membentuk pesan yang berkolisi dengan pesan asli yaitu pesan $1111\ldots1111_H$ (512 bit). Pada $IV = 0000_H$ penyerang dapat membentuk tiga pesan yang berkolisi dengan pesan $1111\ldots1111_H$ (512 bit) yaitu $1111\ldots11000000F2_H$, $1111\ldots110000C70A_H$, dan $1111\ldots110000F4F2_H$.

IV Pseudorandom

Hasil *fixed point* dari penerapan *simplified* IDEA pada skema Davies Meyer dengan input *random* dapat dilihat pada Tabel 2.

Tabel 2. *Hasil Fixed point attack Penerapan simplified IDEA pada Skema Davies Meyer dengan Initial Vector pseudorandom*

IV (4 heksa)	*Plaintext* (136 heksa)	Hasil *Hash*	*Fixed Point* (8 heksa)
3C94	11111....1111	5A7D	00003D3B
	11111....1111		0000E59C
4C04	11111....1111	F519	-
6D33	11111....1111	3EC6	-
9F27	11111....1111	750C	-
24AD	11111....1111	EA5A	-
56E9	11111....1111	3F3D	-
90A4	11111....1111	00DC	00000731
423F	11111....1111	93C2	000063B2
0861	11111....1111	A0EA	-
5043	11111....1111	12E5	-
5817	11111....1111	0451	-
7164	11111....1111	4713	-
7759	11111....1111	D2D7	00002D50
B7DE	11111....1111	FEA1	-
B949	11111....1111	82CD	0000D87F
D9F2	11111....1111	94D1	00009345

Berdasarkan Tabel 2. dapat dilihat bahwa penyerang dapat menemukan *fixed point* dengan jumlah maksimal sebanyak 2 buah dari 65536 variasi *input* pada blok terakhir. Probabilitas yang dibutuhkan untuk memperoleh *fixed point* adalah 0,000031. *Fixed point* yang diperoleh dapat dipergunakan penyerang untuk membentuk pesan-pesan yang berkolisi dengan pesan asli yaitu pesan $1111\ldots1111_H$ (512 bit). Pada $IV = 3C94_H$ penyerang dapat membentuk dua pesan yang berkolisi dengan pesan $1111\ldots11_H$ (512 bit) yaitu $1111\ldots1100003D3B_H$ dan $1111\ldots110000E59C_H$.

3. Kesimpulan

Berdasarkan hasil analisis dengan *fixed point attack* dengan menggunakan sampel ekstrim dan *pseudorandom* dapat disimpulkan bahwa *simplified* IDEA dengan skema Davies-Meyer tidak memiliki sifat *collision resistance* yang baik.

Daftar Pustaka

[1] Merkle, Ralph C, *One Way Hash Functions and DES*, Advances in Cryptology-CRYPTO'89, volume 435, pp 428-446, 1989
[2] Damgard, Ivan B, *A design Principle for Hash Functions*, Advances in Cryptology-CRYPTO'89, volume 435, pp 416-427, 1989.
[3] Chen, Raphael, *New Techniques for Cryptanalysis of Cryptographic Hash Functions*, PhD Thesis, Israel Institute of Technology, 2011.
[4] Bartkewitz, Timo, *Building Hash Functions from Block Cipher, Their Security, and Implementation Properties*, Ruhr-University Bochum, 2009.
[5] Lai, Xuejia; Massey, James L., *Markov Ciphers and Differential Cryptanalysis*, Advances in Cryptology-EUROCRYFT'91, Volume 547, pp 17-38, 1991.
[6] Danda, MK Reddy, *Design and Analysis of Hash Functions*, PhD Thesis, Victoria University, 2007
[7] Preneel, Bart, *Analysis and Design of Cryptographic Hash Functions*, PhD Thesis, Katholieke Universiteit Leuven, 2003.
[8] Hoffman, Nick, *Simplified IDEA Algorithm*, Northern Kentucky Univerity, 2007.
[9] Menezes, Alfred J.; Oorscot, Paul C. Van; Vanston, Scott A., *Handbook of Applied Cryptography*, CRC Press, 1996.

[10] Mouha, N., *Automated Techniques for Hash Function and Block Cipher Cryptanalysis.* Belgium: Khatolieke Universiteit Leuven. 2012
[11] Stinson, D. R., *Cryptography Theory and Practice Third Edition,* USA: Taylor & Francis Grup, LLc,2006

Biodata Penulis

Agus Winarno, memperoleh gelar Sarjana Sain Terapan Teknik Persandian (S.S.T.TP), Jurusan Teknik Persandian Sekolah Tinggi Sandi Negara, lulus tahun 2014.
Syamsi Nurdiansah, memperoleh gelar Sarjana Sain Terapan (S.ST), Jurusan Teknik Persandian Sekolah Tinggi Sandi Negara, lulus tahun 2008. memperoleh gelar Sarjana Teknik (ST), Jurusan Elektro Telekomunikasi Institut Sains dan Teknologi Nasional, lulus tahun 2011.
Sigit Setiono, memperoleh gelar Sarjana Sain Terapan (S.ST), Jurusan Teknik Persandian Sekolah Tinggi Sandi Negara, lulus tahun 2006.

ISSN : 2302-3805