COLLISION RESISTANCE FUNGSI *HASH* BERBASIS *BLOCK CIPHER* DENGAN MENGGUNAKAN ALGORITMA MINI-AES

Kuni Inayah[1)], Bondan Estuwira Sukmono[2)]
[1]Sekolah Tinggi Sandi Negara
Jalan Raya Haji Usa Desa Putat Nutug, Ciseeng, Bogor-Jawa Barat, 16330
HP (penulis utama): +62 857180639 93
E-mail : kuni.kuniinayah.inayah@gmail.com[1)]

Abstrak

Di dalam kriptografi terdapat sebuah fungsi yang sesuai untuk aplikasi keamanan seperti otentikasi dan integritas pesan. Fungsi tersebut dinamakan fungsi hash. Terdapat tiga skema mengkonstruksi fungsi hash berbasis block cipher, salah satunya adalah skema Davies-Meyer. Mini-AES merupakan miniatur atau bentuk sederhana dari algoritma AES yang menjadi stadard saat ini. Penerapan Mini-AES pada skema Davies-Meyer diharapkan dapat mewakili algoritma AES dalam hal ini adalah collision resistance yaitu ketahanannya terhadap suatu serangan kolisi.
Telah dilakukan penelitian terhadap skema Davies-Meyer yang menggunakan algoritma Mini-AES dengan menggunakan uji performa yaitu ketahanannya terhadap Yuval's birthday attack. Dari 120 percobaan yang dilakukan, terdapat 118 buah kolisi dan 46 pasangan input tidak berkolisi dengan nilai modus kolisi yang muncul adalah 1 yaitu sebanyak 43 buah. Nilai tersebut sangat kecil, sehingga skema Fungsi Hash berbasis block cipher menggunakan Mini-AES dapat dikatakan tahan terhadap kolisi.
Kata Kunci :*Davies-Meyer, Mini-AES, Collision Resistance, Yuval's Birthday Attack*

1. PENDAHULUAN

Skema Davies-Meyer merupakan salah satu dari tiga skema fungsi *hash* berbasis *block cipher*. Penerapan Mini AES pada fungsi enkripsi skema Davies-Meyer diharapkan mampu menghasilkan fungsi *hash* berbasis *blockcipher* yang baik. Salah satu syarat keamanan fungsi *hash* adalah *Collision Resistance.* Untuk mengujinya, maka dilakukan uji kolisi menggunakan Yuval's *Birthday Attack.*

1.1. Latar Belakang

Peradaban manusia kini memasuki peradaban informasi sehingga setiap orang memiliki kemudahan dan kebebasan dalam mengakses informasi di berbagai belahan dunia. Namun, tidak setiap informasi berhak diketahui oleh setiap orang. Terdapat informasi-informasi yang hanya boleh diakses oleh pihak tertentu saja. Dengan demikian diperlukan pengamanan terhadap informasi tersebut.
Salah satu upaya mengamankan informasi adalah dengan menerapkan kriptografi. Menurut taksonominya, kriptografi terbagi menjadi tiga kategori yaitu sistem kripto simetris, sistem kripto asimetris, dan fungsi *hash*[3].Fungsi *hash* digunakan pada konteks yang lebih umum untuk menggantikan perlindungan dari integritas dari suatu data yang berukuran besar menjadi *string* yang lebih pendek dengan panjang tetap (*m* bit) sebagai *outputhash*. Untuk proses pengimplementasian yang lebih efisien pada sistem baik perangkat lunak maupun keras, maka dibangunlah suatu fungsi *hash* yang berdasarkan *block cipher*.
Terdapat tiga skema fungsi *hash* berbasis *blockcipher*, salah satunya skema Davies-Meyer. Dalam paper ini menerapkan algoritma Mini-AES sebagai fungsi enkripsi pada skema Davies-Meyer. Penggunaan algoritma Mini-AES yang merupakan miniatur dari AES diharapkan dapat mewakili AES dari segi keamanannnya. Untuk menguji ketahanan fungsi *hash* berbasis *block cipher* yang dihasilkan, dilakukan uji performa dalam hal ini ketahanannya terhadap kolisi atau *collision resistance* dengan menggunakan Yuval's *Birthday Attack.*

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan permasalahan pada penelitian ini adalah sebagai berikut:
1) Bagaimanakah ketahanan skema Fungsi *Hash* berbasis *block cipher* menggunakan algotima Mini-AES terhadap Yuval's *Birthday Attack?*

1.3 Pembatasan Masalah

Pada paperini, terdapat beberapa pembatasan masalah dari rumusan permasalahan di atas, yaitu:
1) Pada penelitian ini, akan dilakukan penerapan Yuval's *Birthday Attack* pada Fungsi *hash* berbasis *block cipher* dengan menggunakan algoritma Mini AES

2) Percobaan dilakukan hanya dengan menggunakan modifikasi minor seragam pada *input* ekstrim.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui ketahanan skema Davies-Meyer yang menggunakan algoritma Mini-AES terhadap Yuval's *Birthday Attack.*

2. LANDASAN TEORI

2.1 Fungsi *Hash*

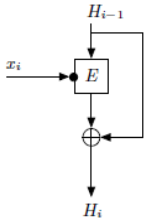
Fungsi *hash* merupakan fungsi yang memetakan *input* dengan panjang sembarang dan menghasilkan *output* dengan panjang tetap. Lebih jelasnya, fungsi *hashh* memetakan *input* bit string dengan panjang sembarang terbatas ke dalam *output* dengan panjang tetap, misal *n*-bit. Fungsi *hash* mempunyai sifat-sifat dasar yaitu kompresi dan mudah dihitung[3].

Terdapat dua klasifikasi fungsi *hash* yaitu *Modification Detection Code* (MDC) dan *Message Authentication Code* (MAC)[7]. Pada klasifikasi MDC, selain dua sifat dasar, terdapat tiga sifat tambahan yaitu *Preimage Resistance, 2nd Preimage Resistance, dan Collision Resistance*[3]. Jika pada suatu fungsi *hash*, kolisi tidak dapat ditemukan dengan mudah *(efficiently solved)* maka fungsi *hash* tersebut dikatakan memiliki sifat *collision resistance*[7].

2.2 Skema Davies-Meyer

Konstruksi fungsi *hash* berbasis *blockcipher* didasarkan pada keefisienannya pada proses implementasi baik pada perangkat lunak maupun perangkat keras. Terdapat tiga skema untuk mengkonstruksi fungsi *hash*berbasis*blockcipher*. Tiga skema tersebut meliputi *Matyas-Meyer-Oseas, Davies-Meyer*,dan *Miyaguchi-Preneel*. Dari ketiga skema yang disebutkan, dalam penelitian ini skema yang digunakan adalah skema Davies-Meyer.

Berikut (gambar 1) merupakan skema Davies Meyer:

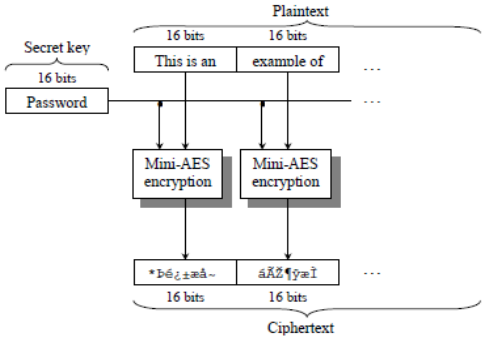


Gambar 1. Skema Davies-Meyer

1) *Input* x dibagi menjadi k-bit *block* dimana k adalah ukuran kunci dan padding jika perlu untuk melengkapi *block* terakhir. Pesan terdiri dari t, k-bit *block*, $x_1,x_2,....,x_t$. IV tetap
2) *Output* didefinisikan dengan Ht, dimana
   $H_0 = IV$
   $H_i = E_{xi}(H_{i-1}) \oplus H_{i-1}, 1 \leq i \leq t$

2.3 Mini AES

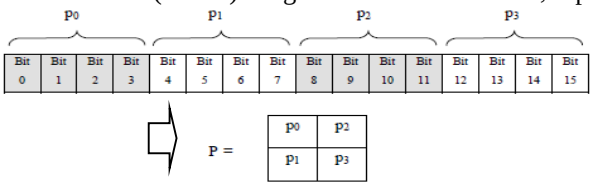
Algoritma Mini-AES adalah miniatur atau bentuk sederhana dari algoritma AES. Untuk mengenkripsi pesan dengan Mini-AES, *input* asli pesan disebut *plaintext* yang kemudian dibagi ke dalam blok-blok, masing-masing blok berukuran 16 bit. Setiap blok *plaintext* dienkripsi menggunakan Mini-AES menjadi *ciphertext*, proses tersebut terus berlangsung hingga semua *plaintext* telah dienkripsi. Enkripsi Mini-AES dilakukan menggunakan kunci rahasia sepanjang 16 bit. Proses tersebut dapat dilihat pada Gambar 2[4].



Gambar 2. Enkripsi Plaintext dengan Mini-AES

2.3.1 Komponen Mini-AES

Agar lebih mudah memahami proses enkripsi Mini-AES, *input* blok *plaintext* 16 bit, P = (p_0, p_1, p_2, p_3) direpresentasikan sebagai matriks 4 bit (*nibble*) dengan 2 baris dan 2 kolom, seperti pada Gambar 3[4].



Gambar 3. Representasi Matriks 2x2 dari Blok 16-bit

Mini-AES mempunyai 4 buah komponen yaitu *NibbleSub, ShiftRow,MixColumn* dan *KeyAddition*.

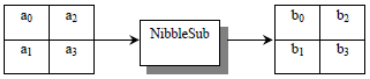
1) *NibbleSub, γ*

NibbleSub adalah operasi sederhana yang mensubstitusikan masing-masing *inputnibble* dengan sebuah *outputnibble* sesuai dengan tabelsubtitusi (*s-box*) 4 x 4 yang ditunjukkan oleh Tabel 1. Nilai dari Tabel 1. ini diambil dari baris pertama *s-box* DES.

Tabel 1. *S-box* Mini-AES

x	0	1	2	3	4	5	6	7	8	9	A	B	C		D	E	F
S(x)	E	4	D	1	2	F	B	8	3	A	6	C	5		9	0	7

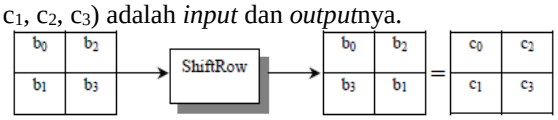
Operasi *NibbleSub* diilustrasikan pada Gambar 4, dimana A = (a_0, a_1, a_2, a_3) adalah *input* blok dan B = (b_0, b_1, b_2, b_3) adalah *output*nya.



Gambar 4. Operasi NibbleSub

2) *ShiftRow, π*

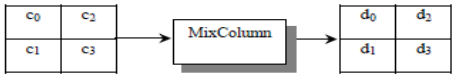
ShiftRow merotasi masing-masing baris dari blok *input* ke kiri oleh sejumlah *nibble* berbeda. Baris pertama tidak berubah dan baris kedua dirotasi ke kiri satu *nibble*. Hal ini diilustrasikan di Gambar 5, dimana B = (b_0, b_1, b_2, b_3) dan C = (c_0, c_1, c_2, c_3) adalah *input* dan *output*nya.



Gambar 5. Operasi ShiftRow

3) *MixColumn, θ*

MixColumn mengambil masing-masing kolom dari blok *input* dan mengalikannya dengan sebuah konstanta matriks untuk memperoleh *output* kolom baru, seperti pada Gambar 6. C = (c_0, c_1, c_2, c_3) dan D = (d_0, d_1, d_2, d_3) menunjukkan *input* dan *output* berturut-turut.



Gambar 6. Operasi MixColumn

dimana $\begin{bmatrix} d_0 \\ d_1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$ dan $\begin{bmatrix} d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \end{bmatrix}$. Perkalian matriks pada fungsi *MixColumn* merupakan perkalian modulus. Modulus yang dipilih pada fungsi*MixColumn* adalah:$m(x) = x^4 + x + 1$

4) *KeyAddition,* σ_{K_i}

KeyAddition menyebabkan masing-masing bit dari blok *input* D = (d_0, d_1, d_2, d_3) di-XOR dengan bit ke-i dari kunci *round* yang berkorespondensi yaitu K_i = (k_0, k_1, k_2, k_3), untuk memperoleh 16 bit blok *output* E = (e_0, e_1, e_2, e_3) seperti pada Gambar 7. Kunci *round* diperoleh dari kunci rahasia K, menggunakan algoritma *key schedule*.

Gambar 7. Operasi KeyAddition

2.3.2 Key-Schedule Mini-AES

Pada Mini-AES, 16 bit kunci rahasia akan melalui *key schedule* untuk memproduksi 16-bit kunci *round*. K_0 digunakan lebih dahulu sebelum *round* pertama, dan sebuah 16-bit kunci *round* K_i, digunakan di masing-masing *round* Mini-AES. Misal sebuah kunci rahasia K berukuran 16 bit didefinisikan sebagai 4 nibble K

= (k_0, k_1, k_2, k_3) untuk memperoleh kunci *round* K_0, K_1, K_2. Begitu juga dengan K_0, K_1, K_2 didefinisikan sebagai 4 *nibble* yaitu $K_0 = (w_0, w_1, w_2, w_3)$, $K_1 = (w_4, w_5, w_6, w_7)$ dan $K_2 = (w_8, w_9, w_{10}, w_{11})$. Nilai kunci *round* diperoleh dari kunci rahasia seperti pada Tabel 2[4]. Catatan pada masing-masing *round*, konstanta *round* rcon(i) digunakan dengan rcon(1) = 0001 dan rcon(2) = 0010.

Tabel 2 Pembangkitan Kunci *Round* Mini-AES

Round	Nilai Kunci *Round*
0	$w_0 = k_0$ $w_1 = k_1$ $w_2 = k_2$ $w_3 = k_3$
1	$w_4 = w_0 \oplus NibbleSub\ (w_3) \oplus$ rcon (1) $w_5 = w_1 \oplus w_4$ $w_6 = w_2 \oplus w_5$ $w_7 = w_3 \oplus w_6$
2	$w_8 = w_4 \oplus NibbleSub\ (w_7) \oplus$ rcon (2) $w_9 = w_5 \oplus w_8$ $w_{10} = w_6 \oplus w_9$ $w_{11} = w_7 \oplus w_{10}$

2.3.3 Enkripsi Mini-AES

Aplikasi dari empat komponen *NibbleSub, ShiftRow, MixColumn* dan *KeyAddition* membentuk satu *round*. Enkripsi Mini-AES dapat dinotasikan oleh :

Mini-$AES_{Encrypt} = \sigma_{K_2} \circ \pi \circ \gamma \circ \sigma_{K_1} \circ \theta \circ \pi \circ \gamma \circ \sigma_{K_0}$[4].

Dimana $\circ$ menunjukkan komposisi fungsi dan susunan eksekusi dari kanan ke kiri, yang berarti σ_{K_0} dilakukan pertama kali.

2.3.4 Dekripsi Mini-AES

Untuk memperoleh *plaintext* asli, kebalikan proses dari enkripsi harus dilakukan pada *ciphertext*. Hal ini disebut dekripsi. Catatan dekripsi adalah invers dari enkripsi.

Mini-$AES_{Decrypt}$ = $(\sigma_{K_2} \circ \pi \circ \gamma \circ \sigma_{K_1} \circ \theta \circ \pi \circ \gamma \circ \sigma_{K_0})^{-1} = \sigma_{K_0} \circ \gamma^{-1} \circ \pi \circ \theta \circ \sigma_{K_1} \circ \gamma^{-1} \circ \pi \circ \sigma_{K_2}$[4].

Untuk proses dekripsi hanya proses *NibbleSub* yang sedikit berbeda, ia membutuhkan tabel *s-box* invers hasil kebalikan dari tabel *s-box* Mini-AES yang ditunjukkan oleh Tabel 3. Sedangkan invers untuk *KeyAddition, ShiftRow* dan *MixColumn* prosesnya sama dengan ketika enkripsi.

Tabel 3 *S-box* invers Mini-AES

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	E	3	4	8	1	C	A	F	7	D	9	6	B	2	0	5

2.4 Yuval's Birthday Attack

Ide dibalik *birthday attack* adalah diketahui bahwa suatu grup berisi 23 orang, probabilitas terdapat minimal dua orang mempunyai hari ulang tahun yang sama adalah lebih dari 0,5. Jumlah 23 orang merupakan jumlah yang sangat kecil yang merupakan perkiraan yang biasa disebut dengan *birthday paradox* [5]. Salah satu contoh penerapan *birthday attack* terdapat dalam *paper* [6] yaitu menemukan kolisi pada MD5. Dalam *paper* tersebut dijelaskan *birthday attack* digunakan untuk meminimalisir percobaan dalam mencari *input* yang berkolisi. Jadi, tujuan *birthday attack* adalah mempermudah menemukan kolisi suatu fungsi *hash*.

Berikut algoritma Yuval's *Birthday Attack* [3]:

Input : Pesan asli x_1, pesan palsu x_2,dan *m*-bit fungsi *hashh.*

Output : x_1', x_2' hasil modifikasi minor dari x_1 dan x_2 dimana $h(x_1') = h(x_2')$.

Langkah-langkah :

1) Bangkitkan $t = 2^{m/2}$ modifikasi minor x_1' dari x_1.
2) Lakukan *hashing* terhadap semua modifikasi minor x_1' dan simpan nilai *hash*nya.
3) Bangkitkan modifikasi minor x_2' dari x_2. Lakukan *hashing* terhadap nilai x_2' dan periksa apakah terdapat nilai yang sama dengan x_1'. Waktu untuk menemukan kolisi melalui *table lookup* adalah konstan dan kolisi ditemukan maksimal setelah *t* kandidat x_2'.

Yuval's *Birthday Attack* tidak hanya digunakan untuk mencari kolisi, tetapi juga untuk menegtahui performa fungsi *hash* dalam hal ini ketahanan kolisi atau *Birthday Attack*.[3].

3. METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode eksperimen dan kajian kepustakaan. Penelitian diawali dengan studi literatur yang diambil dari buku, buku elektronik maupun sumber-sumber yang berhubungan dengan penelitian. Kemudian dilakukan eksperimen dengan mengimplementasikan ke dalam bahasa pemrograman untuk mempermudah perhitungan.

3.2 Tahapan Penelitian

Tahapan dalam penelitian ini adalah:

1) Studi literatur tentangSkema Davies-Meyer, Algoritma Mini-AES, dan konsep fungsi *hash*, skema Davies-Meyer, algoritma Mini-AES dan Yuval's birthday attack.
2) Menerapakan algoritma Mini-AES pada skema Davies-Meyer, mengimplementasikan algoritma Mini-AES pada skema Davies-Meyer ke dalam bahasa pemrograman C++ dan melakukan Yuval's *birthday attack* terhadap skema Davies-Meyer yang menggunakan Mini-AES.
3) Menganalisis kolisi pada skema algoritma Davies-Meyer yang menggunakan Mini-AES.
4) Menampilkan hasil analisis untuk menjawab permasalahan penelitian.

3.3 Populasi dan Sampel

3.3.1 Populasi pengujian

Skema algoritma Davies-Meyer dengan menggunakan algoritma Mini-AES memproses *input* sebesar 16 bit tiap waktu dengan total populasi sebanyak 2^{16} buah.

3.3.2 Sampel dan Teknik Pengambilan Sampel

Pada Yuval's *birthday attack*, memerlukan *input* berupa modifikasi minor dari pesan asli sebanyak $2^{n/2}$ dan pesan palsu sebanyak $2^{n/2}$ dengan *n* adalah bit *output* fungsi *hash*.. Sampel *input* yang digunakan meliputi *input* dengan nilai seragam 0 hingga seragam 1 yang terdiri dari nilai heksa 0 hingga f. Modifikasi minor dilakukan dengan mengubah 8 bit (2 nilai heksadesimal) terakhir paling kanan dari pesan sebanyak 256 buah. Modifikasi minor yang dilakukan hanya menggunakan pasangan *input* ekstrim dengan nilai seragam.

3.4 Teknik Pengumpulan Data

Pengumpulan data dalam penelitian ini mengacu pada data hasil kolisi menggunakan Yuval's *birthday attack.* Untuk menghasilkan data hasil kolisi, dilakukan pembangkitan *input* modifikasi minor sebanyak $2^{n/2}$ yaitu 256 buah sampel. Kemudian dilakukan *hashing* terhadap semua modifikasi minor tersebut dan menyimpan nilai *hash*nya. Kemudian membandingkan hasil *hash* dari modifikasi minor pesan asli dan palsu untuk mencari jumlah kolisi yang terjadi.

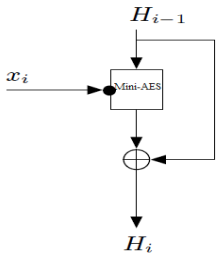
3.5 Teknik Pengolahan dan Analisis Data

Data diolah menggunakan *compiler* pemrograman C++ yaitu program Dev C++ versi 5.2.0.0 dan sebuah laptop dengan spesifikasi: *Processor* Intel Core i7 2.30 GHz dengan *Random Access Memory* (RAM) sebesar 4 *Gigabyte* untuk menjalankan simulasi Yuval's *birthday attack*. Teknik analisis data yang digunakan adalah *performance analysis with attack resistance* yaitu uji performa dalam hal ini adalah ketahanannya terhadap serangan kolisi[2].

4. PENERAPAN YUVAL'S *BIRTHDAY ATTACK* PADA SKEMA DAVIES-MEYER DENGAN MENGGUNAKAN ALGORITMA MINI AES

4.1 Skema Davies-Meyer dengan menggunakan algoritma Mini AES

Algoritma Mini-AES merupakan miniatur dari algoritma AES yang hingga saat ini masih terbukti aman digunakan. Oleh karena itu, penerapan algoritma Mini-AES pada skema fungsi *hash* berbasis Blok *Cipher* Davies-Meyer diharapkan membentuk skema yang aman, dalam hal ini ketahanannya terhadap kolisi.
Berikut merupakan skema Davies-Meyer yang menggunakan algoritma Mini-AES pada proses enkripsinya.



Gambar 8. Skema Mini AES Davies-Meyer

Keterangan :

1) Pada skema Davies-Meyer, kunci block cipher digunakan sebagai input *hash*
2) Menggunakan panjang input 16x5 = 80 bit (total panjang kuncinya adalah 80 bit dengan dioperasikan per 16 bit)
3) Kunci akan dijadikan sampel penelitian ketahanan skema Davies-Meyer menggunakan Mini-AES yaitu berjumlah 256 buah sampel.

4.2 Yuval's *Birthday Attack* pada skema Davies-Meyer dengan Menggunakan Algoritma Mini AES

Penelitian dilakukan dengan mencoba menemukan kolisi dengan mencoba semua kemungkinan pasangan *input-input* ekstrim, yaitu semua pasangan *input* dengan nilai seragam 0 hingga seragam 1 yang meliputi nilai heksa 0 hingga f.Total percobaan yang dilakukan adalah 120 percobaan yang merupakan kombinasi16 kemungkinan *input* berpasangan (C_2^{16} =120). Berikut tabel nilai kolisi dan pasangan *input* yang berkolisi:

Tabel 4.Hasil kolisi dan pasangan input yang berkolisi

Input	***Input* yang berkolisi**		**Kolisi (5 *hexa*)**	**Jumlah kolisi**
	x_1' (100 *hexa*)	**x_2' (100 *hexa*)**		
0-1	000...04c	111...1a8	C56a	2
	000...06a	111...1d1	14d2	
0-3	000...07b	333...383	3f0c	1
0-4	000...058	444...4ed	354c	3
	000...068	444...4f8	ea01	
	000...0ff	444...469	fdea	
0-5	000...01c	555...53b	f0f4	4
	000...07d	555...54c	ead7	
	000...04d	555...51c	26c7	
	000...099	555...587	9e90	
0-6	-	-	-	-
0-7	000...0f4	777...73f	8b38	1
0-8	-	-	-	-
0-9	000...0f0	999...962	2d8a	1
0-a	000...0af	aaa...ac8	a3f3	1
0-b	000...069	bbb...b78	8869	1
0-c	000...04a	ccc...cba	9017	2
	000...064	ccc...ced	0fb0	
0-d	000...066	ddd...d5b	0e28	2
	000...077	ddd...d1e	0729	
0-e	000...0f2	eee...e54	c288	1
0-f	-	-	-	-
1-2	-	-	-	-
1-3	111...154	333...368	2646	1
1-4	111...107	444...424	9ba3	4
	111...151	444...4c8	74af	
	111...161	444...480	ac33	

	111...190	444...47e	0696	
1-5	111...156	555...54d	b29b	1
1-6	111...140	666...616	960c	1
1-7	-	-	-	-
1-8	-	-	-	-
1-9	111...1f8	999...961	39d3	1
1-a	111...1d0	aaa...a5c	aa10	1
1-b	111...10c	bbb...b4b	3405	1
1-c	111...148	ccc...c32	fc9b	1
1-d	111...1a2	ddd...d34	87bf	1
1-e	-	-	-	-
1-f	-	-	-	-
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
a-b	aaa...a51	bbb...be1	4b2f	2
	aaa...ab2	bbb...b3c	89d1	
a-c	aaa...a08	ccc...c2e	e612	2
	aaa...acd	ccc...c0b	b2e6	
a-d	aaa...ab2	ddd...d51	89d1	1
a-e	aaa...a31	eee...e22	5b11	1
a-f	-	-	-	-
b-c	-	-	-	-
b-d	bbb...b3c	ddd...d51	89d1	3
	bbb...b6d	ddd...d6d	392f	
	bbb...b7b	ddd...d8b	63ef	
b-e	bbb...b5d	eee...e0a	7217	1
b-f	bbb...b08	fff...fc4	eb1a	2
	bbb...ba8	fff...fe2	d5fb	
c-d	-	-	-	-
c-e	ccc...c15	eee...ec8	d87d	2
	ccc...cda	eee...e18	432d	
c-f	-	-	-	-
d-e	ddd...d7d	eee...e34	e2e8	1
d-f	-	-	-	-
e-f	eee...eec	fff...f17	9782	2
	eee...e8e	fff...f96	0890	

Melalui data tersebut, diperoleh rata-rata kolisi sebanyak 0,98333 ≈ 0,984 buah dari 120 percobaan. Nilai tersebut merupakan nilai yang sangat kecil, sehingga dapat dikatakan bahwa Fungsi *hash* berbasis block cipher dengan skema Davies-Meyer menggunakan algoritma Mini-AES tahan terhadap Yuval's *birthday attack.*

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan data hasil penelitian, eksperimen dan analisis data, dapat ditarik simpulan hasil penelitian yaitu kolisi yang muncul rata-rata sebanyak 0,98333 ≈ 0,984 buah dengan modus kolisi yaitu 1. Nilai tersebut

sangat kecil sehingga dapat disimpulkan bahwa fungsi *hash*yang dikonstruksi dari skema Davies-Meyer menggunakan algoritma Mini-AES memiliki ketahanan yang cukup baik terhadap Yuval's *birthday attack*[1].

5.2 Saran

Perlu dilakukan pengujian dengan variasi *input-input* yang lain, sehingga perlu diadakan penelitian lebih lanjut mengenai *coliision resistance* dengan menggunakan sampel yang lebih variatif.

6. DAFTAR PUSTAKA

[1] Admi, Adrian. 2012. *Penerapan Yuval's Birthday Attack dan Analisis Kolisi pada Simplified Chaos Hash Algorithm-1 (SCHA-1).* Sekolah Tinggi Sandi Negara: Bogor

[2] Kocarev et al, 2011. *Chaos-Based Cryptography.* Springer-Verlag: Berlin Heidelberg.

[3] Menezes, Alfred J., Paul C. Van Oorschot, Scott A. Vanstone. 1997. *Handbook of Applied Cryptography*. CRC press LLC: Boca Raton.

[4] Phan, Raphael. 2002. *Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students.* Cryptologia XXVI (4).

[5] Preneel, Bart. 2003. *Analysis and Design of Cryptographic Hash Functions*. Belgian National Science Foundation: Belgia.

[6] Rosadi, Febrian Aris. 2007. *Analisis Birthday Attack untuk Menemukan Collision pada Algoritma Hash MD5*. Program Studi Teknik Informatika Intitut Teknologi Bandung: Bandung.

[7] Stinson, Douglas R. 2002. *Cryptography Theory and Practice, third edition.* Chapman & Hall/CRC.