IMPLEMENTASI FUNGSI HASH DENGAN ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) PADA MIKROKONTROLER LPC1769

Syamsi Nurdiansah[1]
[1] *Badan Siber Dan Sandi Negara*
Jl. Harsono RM. No. 70 Ragunan Pasar Minggu Jakarta Selatan 12550
syamsi.nurdiansah@bssn.go.id

Abstrak. Integritas dan otentikasi data merupakan hal penting pada teknik kriptografi. Implementasi kriptografi dalam *embedded system* menjadi kebutuhan pengamanan informasi saat ini. Fungsi Hash merupakan Teknik kriptografi sebagai solusi integritas dan otentikasi data. Pemanfaatan algoritma enkripsi sebagai fungsi hash merupakan langkah efisien dalam implementasi teknik kriptografi. Tulisan ini menjelaskan teknik pemilihan skema implementasi algoritma AES sebagai fungsi hash dengan metode Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO), Miyaguchi – Preneel (MP) sebagai *single-length Modification Detection Code* (MDCs) *of Rate* 1 dan membandingkan kecepatan hasil implementasi pada *embedded system* LPC1769.

1 Pendahuluan

Sistem yang semakin kecil dengan fitur yang semakin canggih adalah teknologi yang diinginkan pengguna saat ini. Guna memenuhi kebutuhan tersebut, perkembangan teknologi *embedded system* melaju dengan pesat. Perkembangan teknologi *embedded system* tidak terlepas dari perkembangan keamanan data dalam sistem tersebut. Teknik Kriptografi merupakan solusi dari pengaman data pada *embedded system*. Sehingga dalam perkembangnya teknik kriptografi selain mengutamakan keamanan juga memperhitungkan platform implementasinya.

Kriptografi merupakan ilmu matematika yang berhubungan dengan aspek pengamanan informasi seperti kerahasiaan (*confidentiality*), integritas data (*data integrity*), otentikasi entitas (*entity authentication*), dan otentikasi keaslian data (*data origin authentication*) [1]. Salah satu perkembangan teknik kriptografi yang bermanfaat guna diimplementasikan dalam *embedded system* adalah pemenuhan kebutuhan aspek kerahasiaan, integritas data dan otentikasi menggunakan satu algoritma. Solusi integritas data dan otentikasi dapat mempergunakan fungsi satu arah (HASH) [1][2]. Metode Davies-Meyer, Matyas-Meyer-Oseas dan Miyaguchi-Preneel merupakan teknik pemanfaatan algoritma enkripsi blok cipher sebagai fungsi hash *single-length Modification Detection Code* (MDCs) *of Rate* 1 [2][3]. AES merupakan algoritma kunci simetrik blok cipher yang dapat di pergunakan untuk memenuhi solusi kerahasiaan data. Dengan Davies-Meyer, Matyas-Meyer-Oseas dan Miyaguchi-

Preneel, AES dapat menjadi solusi memenuhi kebutuhan integritas data dan otentikasi [4].

Dalam tulisan ini dibahas bagaimana melakukan implementasi Davies-Meyer, Matyas-Meyer-Oseas dan Miyaguchi-Preneel dengan skema *single-length Modification Detection Code* (MDCs) *of Rate* 1 menggunakan algoritma AES guna memenuhi kebutuhan integritas data dan otentikasi pada *Integrated Circuit* (IC) LPC-1769. Langkah selanjutnya adalah membandingkan kecepatan satu siklus hash hasil implementasi pada LPC1769.

Dalam proses implementasi fungsi hash mengunakan algoritma AES ini dipergunakan beberapa literatur diantaranya terkait Algoritma AES, LPC-1769, Davies-Meyer, Matyas-Meyer-Oseas dan Miyaguchi-Preneel.

2. Metodologi Penelitian

Penelitian ini dilakukan dengan metode kajian kepustakaan dan eksperimen. Teori yang di pelajari dalam penelitian ini diantaranya terkait Algoritma AES, LPC-1769, Metode Davies-Meyer, Matyas-Meyer-Oseas dan Miyaguchi – Preneel. Langkah selanjutnya adalah melakukan percobaan dengan implementasi dalam mikrokontroller LPC1769.

2.1 Tahapan Penelitian

Penelitian ini memiliki beberapa tahapan diantaranya :

1. Pembelajaran teori terkait Algoritma AES, LPC-1769, Metode Davies-Meyer, Matyas-Meyer-Oseas dan Miyaguchi – Preneel
2. Implementasi algoritma AES 128, 192 dan 256 pada LPC1769.
3. Implementasi varian fungsi hash dengan skema Fungsi Hash MDCs Rate 1 yang cocok dengan kecepatan tiap proses dalam satu siklus.
4. Mengevaluasi kecepatan dari setiap implementasi pada LPC1769

2.2 Eksperimen Penelitian

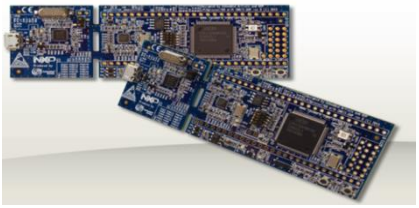
Eksperimen dalam penelitian ini dengan cara mengimplementasikan varian algoritma AES. AES menggunakan jumlah *round* Nr beragam bergantung dari panjang kunci yang digunakan[5]. Tabel 1 merupakan perbandingan jumlah *round* pada AES berdasarkan panjang kunci yang digunakan dimana *Nk words* dan *Nb words* panjang nya 32 bit:

Table 1. Kombinasi Kunci-Blok_Round AES.

Varian AES	*Key Length (Nk words)*	*Block Size (Nb Words)*	*Number Of Rounds (Nr)*
AES 128	4	4	10
AES 192	6	4	12
AES 256	8	4	14

Implementasi algoritma AES dilakukan pada *Board LPCXpresso1769 With CMSIS-DAP* dimana board ini merupakan *board* LPC-1769 yang memiliki akses port debug. Dalam board ini memiliki IC LPC-1769 dimana memiliki kemampuan [6] menggunakan IC NXP ARM Cortex M3, memori SRAM 64 kB, memori Flash 512kB, interface UART 4 buah, interface I2C 3 buah, interface SPI, *Pulse Width Modulation* (PWM), USB 2.0 Device/Host/OTG, USB 2.0 Device/Host/OTG

Bentuk fisik dari *board LPCXpresso1769 With CMSIS-DAP* sebagai berikut :



Gambar 2. *Board LPCXpresso1769 With CMSIS-DAP*

Dalam proses penelitian ini di pergunakan LPCXpresso IDE dalam pembuatan program aplikasinya dan komunikasi UART sebagai interface.

Skema Fungsi Hash MDCs Rate 1 pada varian AES yang diimplementasikan sesuai dengan tabel berikut :

Tabel 1. Fungsi Hash MDc Rate 1 Pada AES

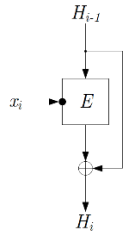
Varian AES	Fungsi Hash MDCs Rate 1	Rate
AES 128	MMO atau MP	n/n = 1
AES 192	DM	k/n
AES 256	DM	k/n

3. Pembahasan Dan Kesimpulan

3.1 Pemilihan Skema Fungsi Hash Untuk AES

a. Metode Davies-Meyer (DM)

Metode ini merupakan metode fungsi kompresi. Skema dari metode DM sesuai dengan gambar berikut [2][3]:



Gambar 3. *Skema Davies-Meyer*

Dimana H_{i-1} dimana H_0 merupakan *Inital Vector* (IV) , x_i merupakan pesan asli yang digunakan seperti kunci masukan pada algoritma E, E merupakan algoritma

enkripsi dan dekripsi blok cipher dan H_i Merupakan nilai hashnya. Untuk nilai hash akhirnya dapat di peroleh dengan rumusan berikut:

$$H_i = E_{x_i}(H_{i-1}) \oplus H_{i-1} \quad \text{.....(1)}$$

untuk $1 \leq i \leq t,\ H_0 = IV.$

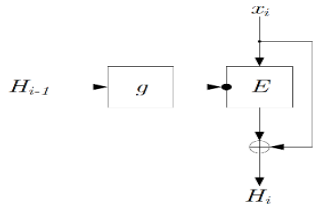
Fungsi hash dengan menggunakan metode DM memiliki nilai rata – rata (*rate*) sesuai dengan rumusan berikut :

$$R_{DM} = \frac{k}{1 \cdot n} = \frac{k}{n} \quad \text{.....(2)}$$

dimana *k* adalah panjang kunci dan n adalah panjang masukan atau keluaran dalam hal ini pesan asli dan pesan terenkripsi.

b. Metode Matyas-Meyer-Oseas (MMO)

Metode MMO memiliki kesamaan dengan metode DM, perbedaanya terletak pada posisi kunci dan pesan asli nya. Bagan metode MMO sebagai berikut [2][3]:



Gambar 4. Skema Metode Matyas-Meyer-Oseas (MMO)

Nilai hash akhir pada skema MMO sesuai dengan rumusan berikut :

$$H_i = E_{g(H_{i-1})}(x_i) \oplus x_i \quad \text{.....(3)}$$

dimana $1 \leq i \leq t,\ H_0 = IV.$

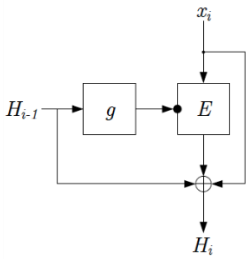
Nilai rata rata (*rate*) dari metode ini

$$R_{MMO} = \frac{n}{1 \cdot n} = 1. \quad \text{.....(4)}$$

Perbedaan yang sangat terlihat dengan metode DM adalah metode MMO ini memiliki panjang kunci dan panjang input output sama.

c. Metode Miyaguchi – Preneel (MP)

Metode kompresi Miyaguchi – Preneel bisa disebut kelanjutan dari metode MMO yaitu dengan menambah kan proses *exclusive-or* (xor) H_i pada keluaran nilai hash nya. Bagan metode MP sebagai berikut [2][3] :



Gambar 5. *Skema Miyaguchi – Preneel* (MP)

Nilai hash akhir pada metode akhir dapat diperoleh dengan rumusan berikut :

$$H_i = E_{g(H_{i-1})}(x_i) \oplus x_i \oplus H_{i-1} \qquad \text{.....(5)}$$

dimana $1 \leq i \leq t$, $H_0 = IV$.

nilai rata – rata (*rate*) dari metode MP adalah

$$R_{MP} = \frac{n}{1 \cdot n} = 1 \qquad \text{.....(6)}$$

Metode MP memiliki kesamaan dengan metode MMO yaitu panjang kunci dengan panjang output sama. AES memiliki tiga varian yaitu AES 128, AES 192 dan AES 256. Untuk AES 128 memiliki input panjang kunci 128 bit dan keluaran 128 bit, untuk AES 192 memiliki panjang kunci 192 bit dan keluaran 128 bit sedangkan untuk AES 256 memiliki panjang kunci 256 bit dan keluaran 128 bit. Sehingga jika di padukan dengan rumus 2, 4 dan 6 maka dapat disimpulkan bahwa metode fungsi hash *single-length Modification Detection Code* (MDCs) *of Rate* 1 pada AES sesuai dengan tabel 1.

3.2 Implementasi Dan Pengujian

Implementasi fungsi hash dengan algoritma AES pada mikrokontroler LPC1769 memiliki dua tahapan yaitu implementasi algoritma AES pada LPC1769 dan implementasi metode fungsi hash *single-length Modification Detection Code* (MDCs) *of Rate* 1 pada AES. Untuk implementasi algoritma AES pada LPC1769 sudah dilakukan pada [8] dengan hasil kecepatan satu siklus sebagai berikut :

Tabel 2. Fungsi Hash MDc Rate 1 Pada AES.

Algoritma	Waktu Proses Enkripsi dan Dekripsi
AES 128	856,92 µs
AES 192	1022,92 µs
AES 256	1202,28 µs

Untuk hasil implementasi fungsi hash *single-length Modification Detection Code* (MDCs) *of Rate* 1 pada AES sesuai dengan tabel berikut :

Table 3. Fungsi Hash MDc Rate 1 Pada AES.

Algoritma	Waktu Proses Satu Siklus Hash
AES-128 MMO	438,66 µs
AES 128 MP	448,86 µs
AES 192 DM	521,66 µs
AES 256 DM	611,34 µs

Dari hasil pengujian pada tabel 3, skema AES-128 MMO memiliki waktu proses satu siklus paling cepat di karenakan jumlah round pada AES 128 paling kecil yaitu 10 round, dibandingkan dengan AES 192 dan AES 256 serta jika dilihat pada gambar 4, skema MMO memiliki skema yang lebih sederhana dibandingkan dengan skema MP pada gambar 5. Waktu terlama adalah skema AES 256 DM dikarenakan memiliki jumlah round paling banyak yaitu 14 round meskipun skema fungsi hash *single-length* MDCs *of Rate* 1 DM hampir sama dengan skema MMO.

4 Kesimpulan

Dalam penelitian ini dapat diambil beberapa kesimpulan diantaranya :

1. Telah berhasil dilakukan pemetaan skema fungsi hash yang sesuai untuk algoritma AES 128, AES 192 dan AES 256
2. Telah berhasil dilakukan implementasi fungsi hash *single-length* MDCs *of Rate* 1 pada AES 128, AES 192, AES 256 dimana waktu tercepat yaitu pada skema AES 128 – MMO 438.66 µs
3. Semakin panjang kunci, semakin banyak round dan semakin banyak proses pada skema fungsi hash maka semakin lama waktu yang di perlukan untuk melakukan satu siklus proses hash.
4. Efisiensi pada sistem embedded untuk memenuhi kebutuhan kemanan data, integritas data dan tentikasi dapat dicapai karena dapat dipenuhi dengan satu algoritma yaitu AES.

Daftar Pustaka

1. Schneier Bruce. *Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C*. California : John Wiley & Sons, Inc 1996
2. J. Menezes Alfred, C. Van Oorschot Paul, A. Vanstone Scott. *Handbook Of Applaid Cryptography*. New York: CRC Press 1997.
3. Bartkewitz Timo "*Building Hash Functions from Block Ciphers, Their Security and Implementation Properties*" Rhur-University Bochum, Februari 23, 2009.
4. Bos. W. Joppe, Ozen Onur, Stam Martijn. "*Efficient Hashing Using the AES Instruction Set*". B. Preneel and T. Takagi (Eds.): CHES 2011, LNCS 6917, pp. 507–522, 2011. _c International Association for Cryptologic Research 2011.
5. NIST,**"**Federal Information Processing Standards Publication (FIPS) 197",*Springfield : National Institute of Standards and Technology (NIST)*, 2001
6. Datasheet LPC1769/68/67/66/65/64/63. Rev. 9.5 — 24 June 2014

7. https://www.embeddedartists.com/products/lpcxpresso/lpc1769_cmsis_xpr.php. Diakses Terakhir Tanggal 4 Januari 2018 Pukul 2.06 WIB
8. Nurdiansah Syamsi, Angraini Novita. "*Implementasi Algoritma Advanced Encryption Standard (AES) pada Mikrokontroler LPC1769*". KNS&I STIKOM BALI 2014.