Algoritma Pembangkit Bilangan Acak Berbasis Fungsi Hash Keccak

Mufi Yanuar Triputranto / 13510106
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
mufi.yanuar@students.itb.ac.id

***Abstrak*—Pembangkit bilangan acak atau disebut juga pseudorandom number generator (PRNG) adalah salah satu aspek penting dalam kriptografi. Makalah ini berisi pemaparan suatu algoritma pembangkit bilangan acak yang berbasis fungsi hash SHA-3 atau biasa dikenal sebagai Keccak. Hasil uji coba membuktikan bahwa Keccak bisa dijadikan basis untuk membuat algoritma pembangkit bilangan acak yang sangat baik selain karena konstruksi sponsnya dan juga fungsi permutasinya yang berbeda dengan algoritma hash lain.**

***Index Terms*—pseudorandom generator, Keccak, spons**

I. PENDAHULUAN

Pembangkit bilangan acak atau disebut juga pseudorandom number generator (PRNG) adalah salah satu aspek penting dalam bidang komputasi. PRNG terutama sangat penting dalam bidang kriptografi, pemodelan dan simulasi, serta game. Bahkan ada juga yang menggunakan PRNG sebagai alat untuk menciptakan seni digital. Deretan angka yang dibangkitkan oleh PRNG ternyata bisa tampak indah bila divisualisasikan dengan cara tertentu.

Fungsi utama dari PRNG adalah—seperti namanya—adalah membangkitkan bilangan acak. Meskipun demikan bilangan tersebut tentu saja tidaklah benar-benar acak (*truly random*). Bilangan *truly random* hanya bisa dibangkitkan lewat hasil pengamatan dari suatu kejadian alam, misalnya peluruhan radioaktif atau perubahan cuaca. Bilangan *truly random* tidak bisa dibangkitkan oleh komputer karena komputer bekerja secara deterministik—bilangan dihasilkan melalui algoritma tertentu yang sifatnya pasti. Oleh karena itulah diberi nama pseudorandom yang berarti acak semu.

PRNG saat ini banyak yang dibuat berdasarkan fungsi hash. Fungsi hash pada dasarnya digunakan untuk menghasilkan digital signature dari suatu file. Namun fungsi hash bisa juga digunakan sebagai basis untuk algoritma PRNG karena memiliki karakteristik melakukan pengacakan terhadap deretan input—yaitu file—dan menghasilkan output berupa signature yang unik. Signature ini bisa diperlakukan seperti layaknya bilangan bertipe integer, long, atau semacamnya. Dari sini bisa disimpulkan bahwa fungsi hash dan PRNG memiliki kesamaan yaitu fungsi satu arah—diberikan suatu bilangan y dan fungsi $f(x)\rightarrow y$, maka hampir tidak mungkin mencari x. Dengan demikian kita bisa membuat suatu PRNG dari fungsi hash apapun selama fungsi hash tersebut memenuhi syarat sebagai fungsi satu arah dan lulus uji statistik.

Salah satu fungsi hash terbaru ialah SHA-3 atau yang biasa dikenal sebagai Keccak. Berbeda dengan fungsi hash sebelumnya, Keccak menggunakan konstruksi spons untuk menghasilkan signature. Dengan konstruksi spons ini Keccak bisa menerima input berupa file dan menghasilkan signature dengan panjang berapapun, berbeda dengan fungsi hash lain yang sifatnya fixed. Keccak sudah diuji dan layak digunakan sebagai fungsi hash.

Pada makalah ini akan dipaparkan suatu algoritma PRNG yang berbasis fungsi hash Keccak. Selain karena tergolong baru, fungsi hash Keccak dipilih karena fungsi permutasinya yang unik.

II. PEMBAHASAN

A. Fungsi Hash Keccak

Prinsip utama fungsi hash Keccak yang membuatnya berbeda dengan fungsi hash lain terletak pada konstruksi spons. Konstruksi spons ini bekerja dalam dua tahap, yaitu tahap absorbsi dan tahap pemerasan (squeezing). Pada tahap absorbsi input berupa blok-blok bitstring diproses ke dalam suatu fungsi permutasi yang memetakan blok bitstring 1 dimensi ke dalam matrix 3 dimensi 5x5xb—dengan b adalah suatu parameter yang menyatakan lebar dari spons—dan melakukan pengacakan. Tahap absorbsi dilakukan sampai semua blok selesai diproses. Pada tahap pemerasan fungsi akan mengembalikan bilangan sepanjang r dengan residu c yang tidak terpakai—perlu diketahui b=r+c. Tahap pemerasan ini bisa dilakukan berulang-ulang. Untuk penjelasan lebih detail bisa dilihat di mengenai konstruksi spons bisa dilihat di [1] dan untuk spesifikasi lengkap Keccak beserta penjabaran matematisnya bisa dilihat di [2].

B. PRNG Keccak

Untuk PRNG dari berbasis Keccak perlu diberikan

beberapa modifikasi. Modifikasi tersebut ialah :

1. Untuk PRNG hanya perlu menerima input seed berupa bilangan, bukan deretan bitstring. Oleh karena itu dibutuhkan penyesuaian agar ada proses absorbsi dalam PRNG. Dalam PRNG yang dipaparkan ini penyesuaian tersebut dilakukan dengan cara memecah bilangan tersebut menjadi sederetan bilangan 8 bit.
2. Untuk PRNG dengan bilangan seed yang pendek perlu dilakukan penyesuaian agar spons tidak kosong saat iterasi pertama pada tahap absorbsi. Penyesuaian dilakukan dengan cara memberikan suatu nilai awal pada spons. Nilai awal yang dipilih pada PRNG ini adalah bilangan natural *e*.

Pseudocode berikut ini adalah hasil modifikasi pada tahap absorbsi dan tahap pemerasan :

```
Absorb(seed,b,limit) :
b : lebar spons/state
A : array 5x5, sebagai spons
e : konstanta awal
r : konstanta penjadwalan
---
bagi seed ke dalam blok 8 bit
if jumlah blok mod 25 != 0 then
    tambahkan padding
for i < jumlah blok do
    for x = {0...4}, y = {0...4} do
        A[x,y] = blok_i ⊕ e[x,y]
        e[x,y] = rotr(e[x,y], r[x,y])
        A = Keccak-f(b,A)
```

```
Squeze(A,b) :
A : spons
b : lebar spons
temp : variabel sementara
---
temp = 0
for x = {0...4}, y = {0...4} do
    temp = temp ⊕ A[x,y] ⊕ e[x,y]
    e[x,y] = rotr(e[x,y],r[x,y])
A = Keccak-f(b,A)
```

Untuk fungsi Keccak-f bisa dilihat di [2] sedangkan rotr adalah fungsi rotate right.

III. Implementasi

Implementasi dilakukan pada lingkungan sistem operasi Windows 7 dengan menggunakan bahasa C#. PRNG bisa diinisialisasi seed terlebih dahulu atau tidak. Untuk nilai awal spons digunakan 475 digit pertama bilangan natural *e* yang dibagi ke dalam matrix 5x5. Hasil keluaran bertipe unsigned long, dengan demikian keluaran dari PRNG nilai maksimal keluaran sebesar 2^{64}-1 dengan rentang antara $0 \leq x \leq 2^{64}$-1. Keluaran bisa dibatasi sampai limit L tertentu saja sehingga hanya menghasilkan bilangan antara $0 \leq x \leq L$.

IV. Uji Coba

Uji coba dilakukan dengan cara mengambil 9999 sampel bilangan random antara 0 sampai 100. Uji coba dilakukan pada PRNG dengan seed dan tanpa seed. Untuk menguji seberapa acak bilangan yang dihasilkan digunakan perhitungan distribusi χ^2 dengan level signifikansi 0.15 dan 100 derajat kebebasan. Berikut adalah hasil uji coba yang didapatkan.

Percobaan	X^2 hasil pengamatan	Batas daerah kritis
Tanpa seed	99.1717	114.6588
Dengan seed	109.2121	
	110.6061	
	117.3333	
	91.63636	
	106.2222	

Dari pengamatan didapat bahwa 1 dari 5 kali percobaan dengan seed melebihi batas kritis. Dengan demikian angka yang dihasilkan PRNG dengan seed berpeluang untuk menghasilkan deret bilangan yang tidak random. Jadi pemilihan seed sangat menentukan deret bilangan yang dihasilkan. Namun jika tingkat signifikansi dikurangi menjadi 0.10 (batas kritis 118.498) maka PRNG bisa dikatakan menghasilkan deret bilangan (pseudo) random, tetapi efeknya ialah PRNG tersebut kurang baik bila digunakan untuk aplikasi kritis

V. Analisis dan Kesimpulan

Dari hasil percobaan dapat ditarik kesimpulan bahwa fungsi hash Keccak bisa dijadikan salah satu basis untuk membuat PRNG. Keunggulan Keccak dibandingkan fungsi hash lain adalah konstruksi sponsnya yang sederhana namun bisa melakukan pengacakan dengan sangat baik. Alasan yang mungkin mengapa hal ini bisa terjadi ialah terletak pada fungsi permutasinya. Pada esensinya operasi permutasi fungsi hash seperti operasi pada ruang vektor : fungsi hash pada umumnya beroperasi pada ruang 1 dimensi sedangkan fungsi Keccak beroperasi pada ruang 3 dimensi. Hal ini mengakibatkan pencarian solusi untuk kombinasi linear lebih sulit. Tentunya hal ini tidak lepas dari operasi xor yang punya karakterisktik mengaburkan informasi. Namun tentunya bisa dilihat bahwa operasi bit xor yang dilakukan pada "3 dimensi" akan lebih kompleks daripada operasi bit xor dalam "1 dimensi". Bila operasi "3 dimensi" ini direduksi menjadi operasi "1 dimensi" yang setara tentunya akan didapatkan bahwa operasi hasil reduksi akan lebih kompleks dibandingkan operasi "1 dimensi" murni.

Secara keseluruhan Keccak sebagai fungsi hash terbaru punya potensi besar untuk digunakan sebagai PRNG yang baik. Seseorang bisa juga membuat PRNG-nya sendiri dengan membuat konstruksi spons seperti Keccak namun dengan fungsi permutasi yang didefinisikan sendiri dan beroperasi dalam "3 dimensi" seperti Keccak.

REFERENSI

[1]. Bertoni, Guido, Joan Daemen, Michael Peeters, dan Gilles Van Assche. "Duplexing the sponge: single-pass authenticated encryption and other applications."

[2]. Bertoni, Guido, Joan Daemen, Michael Peeters, dan Gilles Van Assche. "The Keccak reference."

[3]. Gholipour, A., dan S. Mirzakuchaki. "A Pseudorandom Number Generator with KECCAK Hash." *International Journal of Computer and Electrical Engineering, Vol. 3, No. 6*, 2011.

[4]. National Institute of Standards and Technology. "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications." 2010 (revision 1a).

[5]. Walpole, Ronald E. *Probability & Statistics for Engineers & Scientists*. Pearson, 2012.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

Nama dan NIM