Pengembangan Teknik Pembangkitan Bilangan Acak Berbasiskan Hardware

Yohanes Andika Ruswan Putranto – NIM : 13507067
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
Email : if17067@students.if.itb.ac.id

***Abstract*—Makalah ini akan membahas mengenai proses pembangkitan bilangan acak dengan berbasiskan hardware. Proses pembangkitan bilangan acak merupakan salah satu proses yang paling menentukan dalam proses kriptografi dimana bilangan acak menjadi penentu kekuatan dari kriptografi tersebut. Kebanyakan pembangkitan bilangan acak yang sekarang merupakan pembangkit bilangan acak pseudorandom, oleh karena itu dibutuhkan suatu teknik pembangkitan bilangan acak yang benar-benar acak. Proses dapat dicapai dengan menggunakan hardware sebagai data masukan untuk proses pembangkitan bilangan acak.**

***Index Terms*—pembangkit bilangan acak, kriptografi, hardware, kekuatan kriptografi.**

I. PENDAHULUAN

Bilangan acak merupakan salah satu faktor yang sangat penting dalam kriptografi. Hal ini disebabkan karena bilangan acak menjadi dasar perhitungan dalam kriptografi, yang kemudian menentukan kekuatan dari kriptografi itu sendiri. Misalnya untuk pembangkitan parameter kunci pada algoritma kunci-publik, pembangkitan initialization vector (IV) pada algoritma kunci-simetri, dan sebagainya.

Tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna. Umumnya cara yang digunakan dalam membangkitkan bilangan acak adalah dengan menggunakan basis software, dimana angka-angka yang dibangkitkan berdasarkan suatu algoritma atau fungsi tertentu, sehingga sebenarnya bilangan yang dibangkitkan tersebut bersifat pseudorandom, karena pembangkitan bilangannya dapat diulang kembali.. Apabila bilangan acak yang menjadi dasar dalam kriptografi tersebut bersifat pseudorandom, akan memudahkan bagi kriptanalis untuk memecahkan enkripsi / dekripsi.

Oleh karena itu, dibutuhkan suatu pembangkit bilangan yang benar-benar acak. Bilangan yang benar-benar acak dapat dibangkitkan melalui pengambilan nilai-nilai dari hardware. Seperti, menerima input suara atau foto, kemudian dibangkitkan bilangan berdasarkan suara atau foto tersebut.

II. PSEUDORANDOM NUMBER GENERATOR

Sebagian besar "sumber nomor acak" sebenarnya memanfaatkan pseudorandom generator (PRNG). PRNGs menggunakan proses deterministik untuk menghasilkan serangkaian output dari sebuah negara benih awal. Karena output adalah murni fungsi dari benih data, entropi aktual output dapat tidak pernah melebihi entropi benih. Hal ini dapat, bagaimanapun, layak untuk komputasi membedakan PRNG baik dari RNG sempurna.

Sebagai contoh, sebuah PRNG unggulan dengan 256-bit entropi (atau satu negara dengan 256-bit) tidak dapat memproduksi lebih dari 256 bit yang benar-benar acak benar. Penyerang yang bisa menebak data benih dapat memprediksi seluruh output PRNG. Menebak a 256 - nilai bit benih komputasi dapat dilakukan, namun, ada kemungkinan bahwa sebuah PRNG dapat digunakan dalam aplikasi kriptografi kecil. Contoh PRNGs dirancang untuk kriptografi aplikasi termasuk MD5Random dan SHA1Random (yang masing-masing 128 bit dan 160 bit state) di BSAFETM, sebuah toolkit kriptografi.

Meskipun benar-diimplementasikan dan dipilih benihnya PRNGs paling cocok untuk kriptografi aplikasi, perawatan yang besar harus diambil dalam pengembangan, pengujian, dan pemilihan algoritma PRNG. Sangat penting bahwa PRNG dipilih secara tepat dari sumber yang dapat dipercaya. Sebagai contoh, PRNG yang termasuk dalam library perangkat lunak standar menggunakan benih yang mudah diramalkan atau nilai-nilai state atau menghasilkan output yang dapat dibedakan dari data acak.

Untuk kebutuhan kriptografi, telah banyak terdapat algoritma pseudorandom number generator yang dapat digunakan sebagai proses pembangkitan bilangan acak. Pada bagian ini akan dijelaskan mengenai beberapa algoritma pembangkitan bilangan acak semu yang sudah ada.

A. LCG

Pembangkit bilangan acak kongruen-lanjar (*linear congruential generator* atau *LCG*) adalah *PRNG* yang berbentuk:

$X_n = (aX_{n-1} + b) \bmod m$
X_n = bilangan acak ke-n dari deretnya
X_{n-1} = bilangan acak sebelumnya

a = faktor pengali
b = *increment*
m = modulus

Kunci pembangkit adalah X_0 yang disebut **umpan** (*seed*).

LCG mempunyai periode tidak lebih besar dari m, dan pada kebanyakan kasus periodenya kurang dari itu.

LCG mempunyai periode penuh (m – 1) jika memenuhi syarat berikut:

b relatif prima terhadap m.

a – 1 dapat dibagi dengan semua faktor prima dari m

a – 1 adalah kelipatan 4 jika m adalah kelipatan 4

m > maks(a, b, x0)

a > 0, b > 0

Keunggulan LCG terletak pada kecepatannya dan hanya membutuhkan sedikit operasi bit.

Sayangnya, LCG tidak dapat digunakan untuk kriptografi karena bilangan acaknya dapat diprediksi urutan kemunculannya.

Oleh karena itu LCG tidak aman digunakan untuk kriptografi. Namun demikian, LCG tetap berguna untuk aplikasi non-kriptografi seperti simulasi, sebab LCG mangkus dan memperlihatkan sifat statistik yang bagus dan sangat tepat untuk uji-uji empirik.

Contoh: $X_n = (7X_{n-1} + 11) \bmod 17$, dan $X_0 = 0$

n	X_n
0	0
1	11
2	3
3	15
4	14
5	7
6	9
7	6
8	2
9	8
10	16
11	4
12	5
13	12
14	10
15	13
16	*0*
17	*11*
18	*3*
19	*15*
20	*14*
21	*7*
22	*9*
23	*6*
24	*2*

B. cryptographically secure pseudorandom generator (CSPRNG)

Pembangkit bilangan acak yang cocok untuk kriptografi dinamakan cryptographically secure pseudorandom generator (CSPRNG).

Persyaratan CSPRNG adalah:

- Secara statistik ia mempunyai sifat-sifat yang bagus (yaitu lolos uji keacakan statistik).
- Tahan terhadap serangan (attack) yang serius. Serangan ini bertujuan untuk memprediksi bilangan acak yang dihasilkan.

I. Blum Blum Shut (BBS)

CSPRNG yang paling sederhana dan paling mangkus (secara kompleksitas teoritis).

BBS dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum, dan Michael Shub dan BBS berbasis teori bilangan.

Algoritma:

Pilih dua buah bilangan prima rahasia, p dan q, yang masing-masing kongruen dengan 3 modulo 4.

Kalikan keduanya menjadi n = pq. Bilangan m ini disebut bilangan bulat Blum

Pilih bilangan bulat acak lain, s, sebagai umpan sedemikian sehingga:

(i) 2 □ s < n

(ii) s dan n relatif prima

kemudian hitung x0 = s2 mod n

Barisan bit acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan:

(i) Hitung xi = xi – 1 2 mod n

(ii) zi = bit LSB (Least Significant Bit) dari xi

Barisan bit acak adalah z1, z2, z3, …

Bilangan acak tidak harus 1 bit LSB tetapi bisa juga j buah bit (j adalah bilangan bulat positif yang tidak melebihi log2(log2 n))).

Keamanan BBS terletak pada sulitnya memfaktorkan n. Nilai n tidak perlu rahasia dan dapat diumumkan kepada publik.

BBS tidak dapat diprediksi dari arah kiri (unpredictable to the left) dan tidak dapat diprediksi dari arah kanan (unpredictable to the kanan),

Artinya jika diberikan barisan bit yang dihasilkan oleh BBS, kriptanalis tidak dapat memprediksi barisan bit sebelumnya dan barsian nit sesudahnya

III. HARDWARE RANDOM NUMBER GENERATOR

Dalam komputasi, perangkat keras pembangkit bilangan acak adalah suatu alat yang menghasilkan angka random dari proses fisik. Perangkat seperti ini sering didasarkan pada fenomena mikroskopik seperti kebisingan panas atau efek fotolistrik atau fenomena kuantum lainnya. Proses-proses ini, dalam teori, benar-benar tidak terduga, dan pernyataan teori tentang ketidakpastian tunduk pada uji eksperimental. Sebuah

perangkat keras kuantum berbasis pembangkit bilangan acak biasanya terdiri dari transduser untuk mengubah beberapa aspek dari fenomena fisik menjadi sinyal listrik, amplifier dan sirkuit elektronik lainnya untuk membawa output dari transduser ke dalam dunia makroskopik, dan skema untuk mengkonversi output menjadi representasi digital, seperti angka biner 0 atau 1, yang bervariasi dengan waktu.

Generator nomor acak juga dapat dibangun dari fenomena makroskopik, seperti bermain kartu, dadu, rolet roda dan mesin lotre. Adanya ketidakpastian dalam fenomena ini dapat dibenarkan oleh teori sistem dinamis dan teori chaos tidak stabil. Teori ini menunjukkan bahwa meskipun fenomena makroskopik yang deterministik dalam mekanika Newton, sistem dunia nyata berevolusi dengan cara yang tidak dapat diprediksi dalam praktek karena salah satu perlu mengetahui rincian-mikro kondisi awal dan manipulasi berikutnya atau perubahan.

Meskipun dadu telah banyak digunakan dalam perjudian, dan di akhir-akhir ini sebagai 'elemen mengacak' dalam permainan (misalnya bermain permainan peran), ilmuwan Victoria Francis Galton menjelaskan cara untuk menggunakan dadu untuk secara eksplisit menghasilkan nomor acak untuk tujuan ilmiah, pada tahun 1890.

Hardware nomor acak generator sering relatif lambat, dan mereka mungkin menghasilkan urutan bias (yaitu, beberapa nilai lebih umum dibanding yang lain). Apakah perangkat keras pembangkit bilangan acak cocok untuk aplikasi tertentu bergantung pada rincian baik aplikasi dan generator.

A. Random Number Generator yang sudah ada

Contoh gambar hardware random number generator komersil (dari Portego di Swedia).

I. Generator Generasi Pertama

Generator dari generasi pertama menghasilkan nomor acak oleh dumping bahaya dari sirkuit digital. Mereka terlindung dengan wadah besi ketat frekuensi tinggi untuk memblokir gangguan eksternal. Tanpa perawatan berkelanjutan, kualitas dari bilangan-bilangan acak tidak cukup untuk melewati semua tes statistik. Oleh karena itu perawatan dilakukan oleh driver perangkat, sehingga seperti / dev / random, hardware standar nomor acak generator di Linux dan derivatif Unix lainnya.

Gambar random number generator generasi pertama

Rp1 adalah perangkat port paralel, menghasilkan nomor hardware 5-bit acak berdasarkan bahaya.

Generator ini dapat ditukar ketika panas dan memiliki konsumsi daya yang rendah (sekitar 0,1 W). Perangkat diaktifkan dari port paralel - tidak ada baterai atau kabel.

* Speed: 2 MByte / s (Port Paralel hanya bisa membaca sekitar 1 MByte / s.)
* Dimensi: 6 x 5 x 1,5 cm
* Berat: 55 g

II. Generator Generasi Kedua

Generasi kedua dari generator nomor acak hardware didasarkan pada generasi pertama dan teorema limit sentral

Pendek kata dapat ditunjukkan oleh teorema limit sentral bahwa dengan penambahan modulo yang cukup banyak nomor acak benar panjang memori m dari proses markov secara efektif dikurangi menjadi satu dan probabilitas transisi yang didekati dengan nilai-nilai yang sama. Ini berarti bahwa entropies dan penutup entropi memiliki nilai teori maksimum satu. Hal ini hanya perlu untuk memverifikasi bahwa cukup banyak nomor acak benar digunakan, karena ini berarti bahwa kondisi Lindeberg terpenuhi.

Keuntungan utama adalah bahwa sementara produksi nomor acak yang baru hanya membutuhkan waktu satu siklus clock, angka-angka acak yang lulus semua tes, i. e. 15 diehard tes, tanpa proses selanjutnya dan yang ini dapat ditampilkan eksperimental dan teori. Selain itu tidak perlu ada perlindungan, dan generator paling lambat di sini adalah minimal delapan juta kali lebih cepat dari / dev / random, hardware standar nomor acak generator di Linux dan Unix derivatif lainnya.

Dapat ditunjukkan bahwa generator ini lulus semua uji statistik bahkan di ruang iklim di -10 ... +60 ° C, seperti dalam sertifikat dari Kryptografics.

Generator ini dioptimalkan untuk operasi a) otonom, b) hard real time dan c) kualitas tinggi dengan kecepatan tinggi. Oleh karena itu ini juga dapat digunakan untuk radar kebisingan, radar stealth, sebagai mengganggu

pemancar dan sebagai clock generator acak (dengan durasi siklus terdistribusi diskrit eksponensial).

Gambar random generator generasi kedua

rw2 adalah kartu ISA, menghasilkan angka 16-bit hardware acak berdasarkan teorema limit sentral.

Angka-angka yang dihasilkan pada kecepatan lebih dari 16,67 MByte / s dan dipancarkan pada frekuensi clock (0 .. 8,33 MHz) sebagai gangguan digital (urutan acak bit independen, white noise dengan frekuensi kritis = 1 / 4 frekuensi clock) menggunakan konektor BNC. Kartu ini juga hot swappable dan juga memiliki keluaran stereo (0 .. 2 * 4.17MHz), tiga kontrol LED (tegangan suplai, BNC output, jam) dan dapat digunakan tanpa bus-kartu ISA karena tidak hanya perlu 5V dan sebuah jam.

* Speed (tanpa overclocking): 16,67 MByte / s (Bus ISA hanya bisa membaca sekitar. 2 MByte / s)
* Dimensi (termasuk IBM Bracket): 18 x 12 x 2 cm
* Power Supply: 5V/0.75A
* Berat: 250 g

B. Random Number Generator dengan berdasarkan gambar

Seperti yang kita ketahui setiap gambar memiliki nilai – nilai yang berbeda satu sama lain. Meskipun mata orang awan menyatakan dua gambar adalah sama, namun di dalam komputer, representasi nilai kedua gambar tersebut berbeda.

Yang saya usulkan disini dalam membuat random number generator adalah mengambil nilai – nilai yang terdapat didalam gambar tersebut sebagai nilai yang menjadi number.

Ada banyak cara pengambilan nilai, seperti :

1. Mengambil LSB dari setiap pixel yang ada di dalam setiap baris gambar.

   Dengan menggunakan cara seperti ini, dari sebuah gambar saja kita sudah dapat menemukan banyak sekali nilai bilangan acak yang dapat muncul.

   Setiap pixel memiliki LSB yang umumnya berbeda, sebuah pixel yang menurut mata kita sama saja sebenarnya mungkin memiliki nilai pixel yang berbeda.

2. Mengambil nilai dari pixel tersebut

   Hal ini merupakan cara yang paling mudah. Kita cuma perlu mengambil nilai dari pixel tersebut, dan kita sudah mendapatkan suatu nilai.

   Mengenai pixel mana yang diambil, kita dapat menentukan, penentuan pixel mana yang diambil, sebab pixel manapun yang diambil tetap akan menghasilkan bilangan acak.

   Untuk menambah tingkat keteracakan dari bilangan, kita dapat juga membuat untuk tiap kali pemanggilan fungsi, maka kita menggunakan gambar yang berbeda. Kalau hal ini dilakukan, akan lebih mudah apabila kita menggunakan kamera sebagai alat bantu.

   Jadi gambar langsung diambil saat kita akan memanggil pembangkitan bilangan acak.

Gambar Lena ukuran kecil

Gambar Lena berukuran besar

Dari dua gambar tersebut saja, kita dapat membuat 2 buah random number yang berbeda, meskipun mata kita menyatakan sama.

Perbandingan LSB dari gambar 1 dan gambar 2, dengan urutan pixel dari kiri ke kanan dalam baris.

Pixel	LSB	LSB

1	1	1
2	1	1
3	1	0
4	1	0
5	0	1
6	1	0
7	1	0
8	0	0
9	0	0
10	0	1
11	1	1
12	1	1
13	1	0
14	1	0

Dengan menyatukan LSB yang didapat tersebut, kita dapat menentukan sebuah bilangan acak. Selain itu, kita bisa menentukan sendiri berapa panjang ukuran bit yang kita butuhkan.

Dengan membuat program yang tepat, yakni menerima input dari user, berapa panjang nilai random yang dibutuhkan, maka program akan melakukan pembacaan bit sesuai dengan panjang yang diminta, lalu mengembalikan hasil tersebut ke layar.

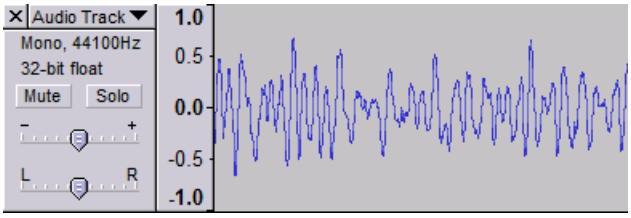
C. Random Number Generator dengan Berdasarkan Suara

Seperti yang sudah kita ketahui, sebuah suara yang tersimpan didalam komputer memiliki banyak sekali nilai – nilai. Nilai – nilai tersebut dapat kita gunakan untuk melakukan pembangkitan nilai acak.

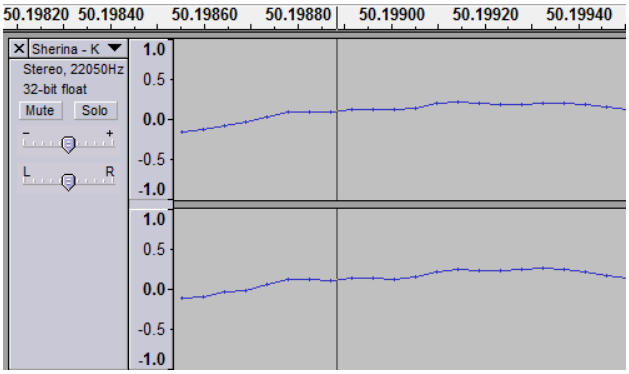
Berbagai cara yang dapat dilakukan dalam mendapatkan nilai acak dari sebuah suara :

1. Mengambil nilai frekuensi dari stream suara dan dihubungkan dengan waktu

Nilai frekuensi dari stream suara umumnya akan selalu berubah (kecuali kalau kita melakukan generate suatu tone, atau mengosongkan suara dalam jangka waktu tertentu melalui komputer). Hal ini membuat kita dapat mengambil suatu nilai dari stream tersebut. Namun karena nilai stream berkisar hanya disekitar nilai tertentu, kita membutuhkan nilai lain yang membuat bilangan yang dihasilkan lebih bervariasi. Oleh karena itu, kita melakukan



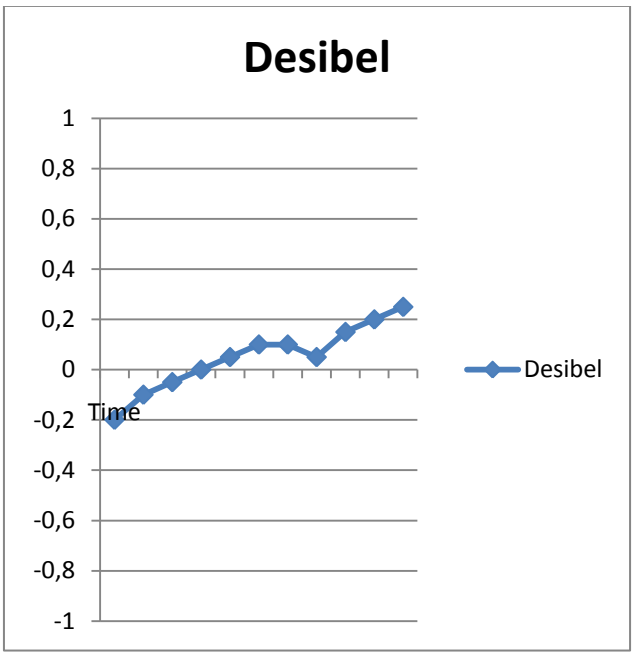
Contoh grafik desibel suara terhadap waktu.



Contoh keterhubungan antara desibel dengan waktu.

Dari gambar diatas, kita dapat melakukan sampling dari frekuensi file tersebut.

Grafik hubungan desibel terhadap waktu frekuensi adalah :



Dengan melakukan encoding dari data tersebut, kita dapat memperoleh nilai bit.

Contoh proses encoding adalah :

Nilai Desibel	Bit yang menjadi output
-1	00000000000000000000000000000000
-0,95	00000000000000000000000000000001
-0,90	00000000000000000000000000000011
-0,85	00000000000000000000000000000111
-0,80	00000000000000000000000000001111
-0,75	00000000000000000000000000011111
-	00000000000000000000000000000000

0,70	00111111
-0,65	0000000000000000000000000000000001111111
-0,60	0000000000000000000000000000000011111111
-0,55	0000000000000000000000000000000111111111
-0,50	0000000000000000000000000000001111111111
-0,45	0000000000000000000000000000011111111111
-0,40	0000000000000000000000000000111111111111
-0,35	0000000000000000000000000001111111111111
-0,30	0000000000000000000000000011111111111111
-0,25	0000000000000000000000000111111111111111
-0,20	0000000000000000000000001111111111111111
-0,15	0000000000000000000000011111111111111111
-0,10	0000000000000000000000111111111111111111
-0,05	0000000000000000000001111111111111111111
0,00	0000000000000000000011111111111111111111
0,05	0000000000000000000111111111111111111111
0,10	0000000000000000001111111111111111111111
0,15	0000000000000000011111111111111111111111
0,20	0000000000000000111111111111111111111111
0,25	0000000000000001111111111111111111111111
0,30	0000000000000011111111111111111111111111
0,35	0000000000000111111111111111111111111111
0,40	0000000000001111111111111111111111111111
0,45	0000000000011111111111111111111111111111
0,50	0000000000111111111111111111111111111111
0,55	0000000001111111111111111111111111111111
0,60	0000000011111111111111111111111111111111
0,65	0000000111111111111111111111111111111111
0,70	0000001111111111111111111111111111111111
0,75	0000011111111111111111111111111111111111
	11111111
0,80	0000111111111111111111111111111111111111
0,85	0001111111111111111111111111111111111111
0,90	0011111111111111111111111111111111111111
0,95	0111111111111111111111111111111111111111
1,00	1111111111111111111111111111111111111111

Lalu, tiap 8 desibel tersebut, jumlahkan 4 bit pertama lalu dikurang dengan 4 bit terakhir, hasil tersebutlah yang menjadi bit output.

V. KESIMPULAN

Random number generator dengan cara mengambil nilai dari gambar atau dari suara tidak membutuhkan alat tambahan, tidak seperti pembangkitan bilangan acak dengan menggunakan entropi transistor.

Random number generator dengan mengambil dari gambar atau dari suara lebih mudah untuk dilakukan.

Pseudorandom generator akan dapat dengan mudah ditebak oleh kriptanalis.

Suara akan lebih mudah menghasilkan random number yang berbeda. Karena suara mudah sekali berubah daripada gambar. Tiap suara yang direkam pastilah memiliki perbedaan.

REFERENSI

Slide kuliah IF3058 Kriptografi

http://true-random.com/

http://www.robertnz.net/hwrng.htm

http://www.robertnz.net/true_rng.html

J. Benjamin and K. Paul, "The Intel Random Number Generator", CRYPTOGRAPHY RESEARCH, INC. WHITE PAPER PREPARED FOR INTEL CORPORATION, April 1999

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, 17 Mei 2010

Yohanes Andika
13507067