IMPLEMENTASI *REAL TIME VOICE SCRAMBLER* PADA DSK TMS320C5402 DENGAN MENGGUNAKAN METODE KRIPTOGRAFI RC4

Mike Yuliana[1], Titon Dutono[1], Wirawan[2]
[1]*Electrical Engineering Polytechnic Institute of Surabaya (EEPIS), Surabaya, Indonesia*
[2]*Institut Teknologi Sepuluh Nopember(ITS), Surabaya, Indonesia*
e-mail: mieke@eepis-its.edu, mikeyuliana@yahoo.com

ABSTRAKSI

Pada paper ini, metode kriptografi RC4 akan diimplementasikan pada DSK TMS320C5402 sebagai metode yang digunakan untuk mengacak suara. Sistem tersebut bisa berhasil bila proses enkripsi/dekripsi bisa real time, dan dari hasil pengukuran terlihat bahwa sistem yang dihasilkan bisa real time karena waktu eksekusi proses enkripsi/dekripsi tiap sampel tidak melebihi 125 μ detik, dimana waktu eksekusi yang dibutuhkan untuk proses enkripsi adalah 1,260 μ detik sedangkan waktu eksekusi yang dibutuhkan untuk proses dekripsi adalah 1,140 μ detik.

Dari hasil analisa terlihat bahwa sinyal suara hasil enkripsi berbeda dengan sinyal suara aslinya, sedangkan sinyal suara hasil dekripsi sama dengan sinyal suara aslinya dan apabila kita dengarkan ternyata sinyal suara hasil dekripsi sama dengan sinyal suara aslinya . Hal ini menunjukkan keberhasilan DSK TMS320C5402 sebagai enkriptor/dekriptor suara, karena suara hasil dekripsi bisa didengarkan seperti sinyal suara aslinya

Kata kunci: *DSK TMS320C5402, Metode enkripsi/dekripsi RC4, MOS, Intelligibility suara.*

1. PENDAHULUAN

Dengan semakin maraknya orang memanfaatkan layanan komunikasi baik melalui jalur telepon maupun jalur internet, maka permasalahanpun bermunculan, apalagi ditambah dengan adanya *hacker* dan *cracker*. Banyak orang kemudian berusaha menyiasati bagaimana cara mengamankan informasi yang dikomunikasikannya, atau menyiasati bagaimana cara mendeteksi keaslian dari informasi yang diterimanya. Oleh karena itulah, dibutuhkan suatu metode kriptografi yang bertujuan untuk menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah.

Algoritma RC4 adalah salah satu jenis *stream cipher* yang dibuat oleh Ron Rivest, pada tahun 1987 untuk perusahaannya, RSA Data Security Inc.[ ]. Pada eksperimen ini akan diuji tingkat keberhasilan DSK TMS320C5402 sebagai enkriptor/dekriptor suara.

Pada paper ini dijelaskan pada bab 2 tentang dasar teori dari metode enkripsi/dekripsi RC4, serta modul DSK TMS320C5402, pada bab 3 dijelaskan tentang implementasi metode kriptografi RC4 pada DSK TMS320C5402, pada bab 4 dijelaskan tentang pengujian dan analisa hasil eksperimen, dan pada bab 5 adalah kesimpulan.

.

2. DASAR TEORI SISTEM

2.1 Metode Enkripsi/Dekripsi RC4

RC4 mempunyai sebuah S-Box, S_0, S_1, ….., S_{255}, yang berisi permutasi dari bilangan 0 sampai 255. Menggunakan dua buah indeks yaitu i dan j di dalam algoritmanya. Indeks i digunakan untuk memastikan bahwa suatu elemen berubah, sedangkan indeks j akan memastikan bahwa suatu elemen berubah secara random. Intinya, dalam algoritma enkripsi metode ini akan membangkitkan *pseudorandom byte* dari *key* yang akan dikenakan operasi XOR terhadap *plaintext* untuk menghasilkan *ciphertext*. Dan untuk menghasilkan *plaintext* semula, maka *ciphertext*-nya akan dikenakan operasi XOR terhadap *pseudorandom byte*-nya.

Secara garis besar algoritma dari metode RC4 ini terbagi menjadi dua bagian, yaitu: *Key Setup* dan *stream generation.* Berikut ini akan dijelaskan *step by step* algoritma metode RC4:

a. *Key Setup*

Pada bagian ini, terdapat tiga tahapan proses yaitu:

1. Inisialisasi S-Box
   Pada tahapan ini, S-Box akan diisi dengan nilai sesuai indeksnya untuk mendapatkan S-Box awal. Inisialisasi S-Box adalah sebagai berikut :
   for i = 0 to 255
   S[i] = i
2. Menyimpan *key* dalam *Key Byte Array*
   Pada tahapan ini, kunci yang akan kita gunakan untuk mengenkripsi atau mendekripsi akan dimasukkan ke dalam *array* berukuran 256 *byte* secara berulang sampai seluruh array terisi.
3. Permutasi pada S-Box
   Pada tahapan ini, akan dibangkitkan sebuah nilai yang akan dijadikan aturan untuk permutasi pada S-Box dengan operasi sebagai berikut:
   j = 0
   for i = 0 to 255
   j = (j + S[i] + K[i]) mod 256
   pertukarkan isi S[i] dan isi S[j]

b. *Stream Generation*

Pada tahapan ini akan dihasilkan nilai *pseudorandom byte* dari *key* yang akan dikenakan operasi XOR untuk menghasilkan *ciphertext* ataupun sebaliknya untuk menghasilkan *plaintext.* Untuk membangkitkan kunci enkripsi dilakukan proses sebagai berikut:

x= y = 0
x = (x+1) mod 256
y = (y + S[x]) mod 256
pertukarkan isi S[i] & S[j]
k = S[(S[x] +S[y]) mod 256]

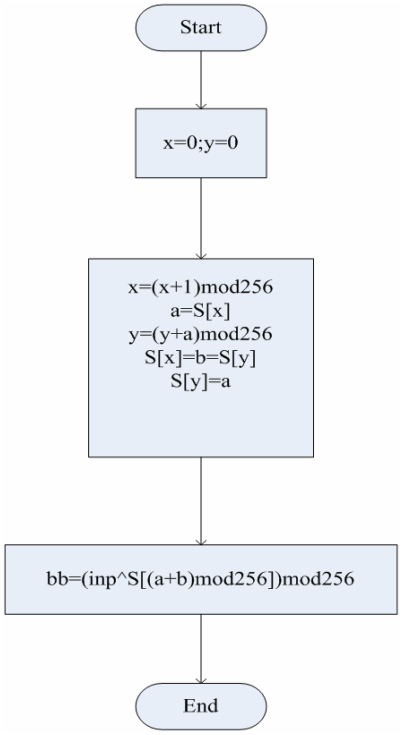
Sedangkan untuk membangkitkan kunci dekripsi akan dilakukan proses sebagai berikut :

xx= y y= 0
xx = (xx+1) mod 256
yy = (yy + S[x]) mod 256
pertukarkan isi S[xx] & S[yy]
k = S[(S[xx] +S[yy]) mod 256]

k merupakan kunci yang langsung beroperasi terhadap *plaintext(P)* ataupun *ciphertext(C),* sedangkan *K* adalah kunci utama atau kunci induk.

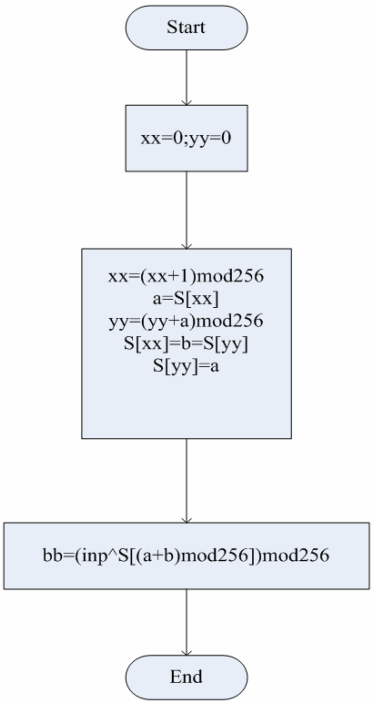
$$C = P \oplus k$$
$$P = k \oplus C$$

Adapun diagram alir dari proses enkripsi adalah sebagai berikut:



Gambar 1. Diagram Alir Proses Enkripsi

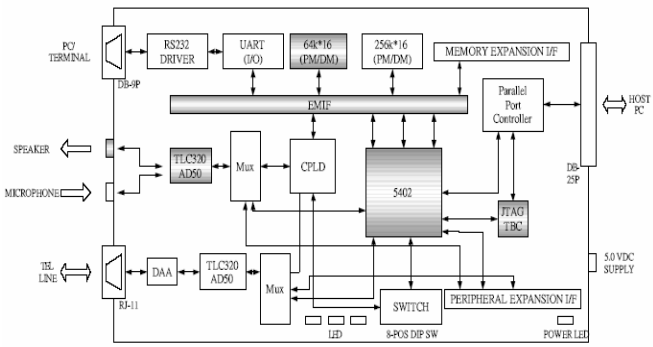
Sedangkan diagram alir dari proses dekripsi adalah sebagai berikut:



Gambar 2. Diagram Alir Proses Dekripsi

2.2 DSK TMS320C5402

Pada penelitian ini, metode enkripsi/dekripsi RC4 diimplementasikan pada DSP *Starter Kit* (DSK) C5402. Adapun blok diagram dari DSK C5402 bisa dilihat pada Gambar 3.



Gambar 3. DSK TMS320C5402

DSP C54x memakai modifikasi lanjut dari arsitektur *Harvard* yang meningkatkan kekuatan proses dengan 8 bus (4 program/data dan 4 address)

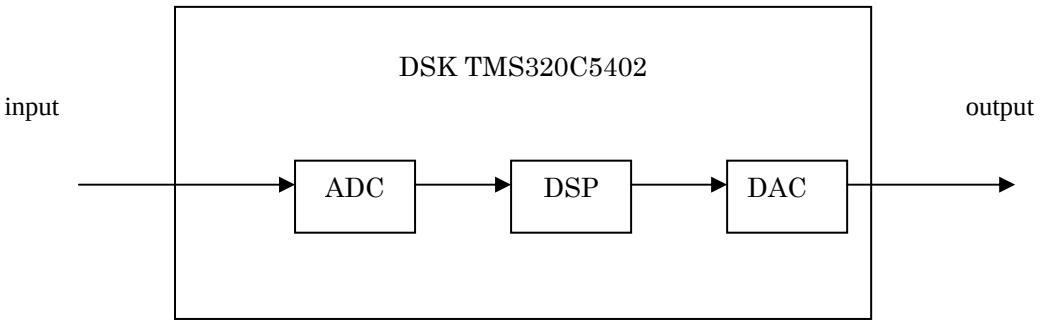
- PB (program bus): membawa *instruction code* dan *immediate operand* dari *program memory*
- Tiga bus data
  - CB (*coefficient bus*): membawa *operand* yang dibaca dari data *memory.*
  - DB (*data bus*): membawa *operand* yang dibaca dari data *memory*.
  - EB (*write bus*): membawa data yang akan ditulis ke *memory*.
- Empat *address bus*: PAB, CAD, DAB, EAB, membawa alamat yang dibutuhkan untuk program *execution*.

Jalur untuk program dan data terpisah membuat akses serentak dari program instruksi dan data, menyediakan mekanisme paralel yang tinggi. Sebagai contoh, tiga proses membaca dan satu proses menulis dapat dilakukan pada satu *cycle*. Instruksi dengan penyimpanan paralel dan instruksi untuk aplikasi khusus sangat memerlukan arsitektur ini. Sebagai tambahan, data dapat ditransfer diantara data dan *program space*. Mekanisme paralel seperti ini mendukung kemampuan untuk aritmatik, logika, operasi manipulasi bit yang semuanya dapat dilakukan pada satu mesin *cycle* saja. Dan juga, C54x mencakup mekanisme kontrol untuk menangani interupsi, operasi pengulangan dan fungsi call. Komponen didalam *Central Processing Unit* (CPU) didukung oleh:

o 40-bit ALU
o dua 40-bit *accumulators*
o 40-bit *Barrel Shifter*
o 17 x 17 *multiplier*
o 40-bit *adder*
o *compare, select, and store unit* (CSSU)
o *Data address generation unit*
o *Program address generation unit*

3. IMPLEMENTASI METODE KRIPTOGRAFI RC4 PADA DSK TMS320C5402

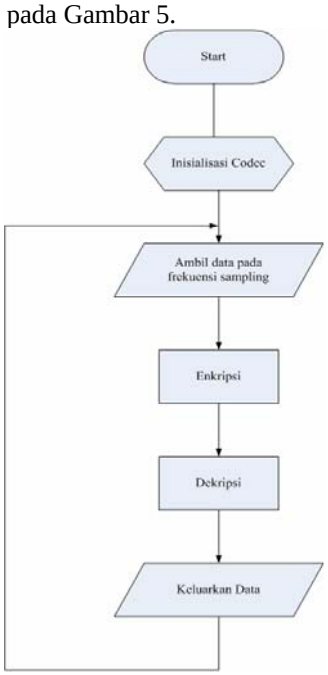
Pada tahap ini, akan dilakukan integrasi metode kriptografi RC4 dengan DSK TMS320C5402, dimana blok diagram sistemnya bisa dilihat pada Gambar 4.



Gambar 4. Blok Diagram Integrasi Metode Enkripsi/Dekripsi RC4 pada DSK TMS320C5402

Pada proses enkripsi/dekripsi suara akan dibutuhkan ADC untuk mengkonversi sinyal analog menjadi sinyal digital serta DAC untuk mengkonversi sinyal digital menjadi sinyal analog.

Adapun diagram alir dari integrasi metode enkripsi/dekripsi RC4 pada DSK TMS320C5402 bisa dilihat pada Gambar 5.



Gambar 5. Diagram Alir Metode Enkripsi/Dekripsi RC4 pada DSK TMS320C5402

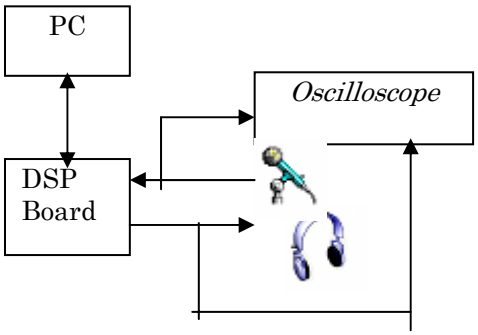
4. PENGUJIAN DAN ANALISA

Pada tahap ini, akan dilakukan pengujian DSK TMS320C5402 sebagai enkriptor/dekriptor suara dimana nantinya bisa dilihat sejauh mana tingkat keberhasilan dari implementasi metode enkripsi/dekripsi RC4 pada DSK TMS320C5402.

Sebelum dilakukan pengujian DSK sebagai enkriptor/dekriptor suara, maka dilakukan persiapan peralatan terlebih dahulu, dimana peralatan yang digunakan ditunjukkan oleh Gambar 6 sedangkan secara diagram ditunjukkan oleh Gambar 7.

Gambar 6. Peralatan yang Digunakan

 ISSN: 1907-5022



Gambar 7. Diagram Peralatan

Adapun spesifikasi peralatan bisa dilihat pada tabel 1.

Tabel 1. Spesifikasi Peralatan

Peralatan	***Spesifikasi***
PC	Intel Pentium 4 2.26 GHz RAM 256 MB OS Windows XP SP2
Pembangkit Sinyal	IWATSU SG-4105 DDS (Direct Digital Synthesis) Fmax 15 MHz, accuracy ±50 ppm. Output max.±10Volt
Oscilloscope	KENWOOD DCS-8300 DC – 20MHz (-3dB) at 1 or 2 mV/division
Board DSP	DSK TMS320C5402

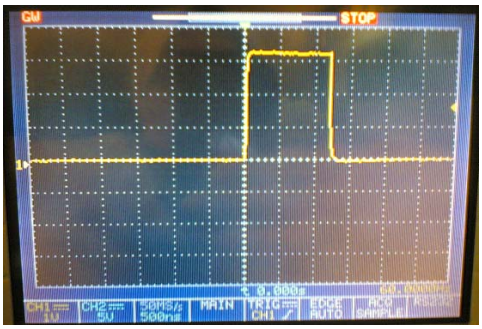
Real time atau tidaknya suatu sistem tergantung dari lama waktu eksekusi yang dibutuhkan untuk proses enkripsi/dekripsi tiap sampel. Pada sistem ini digunakan frekuensi sampling sebesar 8 KHz. sehingga periode sampling adalah sebesar 125 μdetik. Hal ini berarti jarak tiap sampel adalah 125 μdetik. Artinya proses enkripsi/dekripsi satu sampel haruslah selesai sebelum sampel berikutnya muncul, dengan kata lain waktu eksekusi tidak boleh melebihi 125 μdetik. Apabila proses enkripsi/dekripsi satu sampel lebih lama dari jarak tiap sampel maka sistem menjadi tidak *real time* karena ada beberapa sampel yang terbuang (tidak diproses) sehingga proses rekonstruksi sinyal menjadi salah.

Pengukuran waktu eksekusi yang dibutuhkan untuk proses enkripsi/dekripsi satu sampel dilakukan dengan cara men-*toggle* bit XF dimana bit XF ini terhubung dengan pin prosesor DSP. Pin XF merupakan pin output serba guna. Pin XF ini selanjutnya dihubungkan ke *oscilloscope* untuk diukur rentang waktunya.Untuk melakukan *toggle* digunakan instruksi set bit dan reset bit dalam bahasa assembly. Baris program yang akan diukur waktu proses eksekusinya "dibungkus" dengan instruksi berikut.

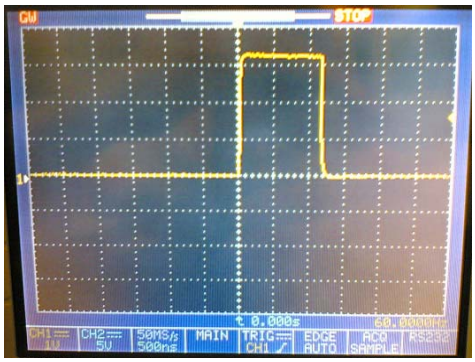
```
asm(" SSBX XF"); // XF=1
...
// fungsi enkripsi/dekripsi RC4
...
asm(" RSBX XF"); // XF=0
```

Dari hasil pengukuran didapatkan lama waktu eksekusi yang dibutuhkan untuk proses enkripsi adalah 1,260 μ detik sedangkan lama waktu eksekusi yang dibutuhkan untuk proses dekripsi adalah 1,140 μ detik. Karena lama waktu eksekusi proses enkripsi/dekripsi tiap sampel tidak melebihi 125 μ detik, maka sistem yang dihasilkan akan bisa *real time.*

Gambar 8 dibawah ini menunjukkan hasil pengukuran sinyal XF untuk lama waktu eksekusi yang dibutuhkan untuk proses enkripsi, sedangkan Gambar 9 menunjukkan hasil pengukuran sinyal XF untuk proses dekripsi.



Gambar 8. Pengukuran Sinyal XF untuk Proses Enkripsi

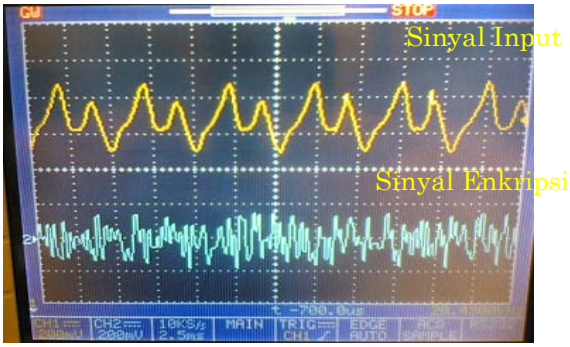


Gambar 9. Pengukuran Sinyal XF untuk Proses Dekripsi

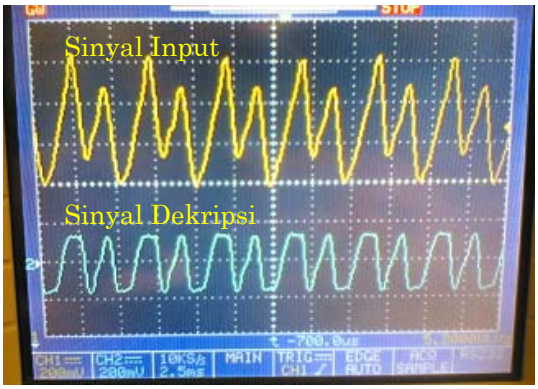
Tingkat keberhasilan DSK TMS320C5402 sebagai enkriptor/dekriptor suara bisa diuji dengan menganalisa sinyal suara serta mendengarkan suara yang dihasilkan. Pengujian sinyal ini dilakukan pada tiga kondisi yaitu Sinyal suara yang merupakan input bagi *codec*, sinyal suara yang telah terenkripsi, serta sinyal suara yang telah didekripsi. Hasil pengamatan pada *oscilloscope* bisa dilihat pada Gambar 10-11.

Dari Gambar 10 dan Gambar 11 terlihat bahwa sinyal enkripsi mempunyai bentuk acak yang tidak sama dengan sinyal input, sedangkan sinyal output mempunyai bentuk yang sama dengan sinyal input namun mengalami penguatan. Dan apabila didengarkan ternyata suara hasil dekripsi sama dengan suara aslinya, hal ini menunjukkan keberhasilan DSK TMS320C5402 sebagai enkriptor/dekriptor suara.

 ISSN: 1907-5022



Gambar 10. Sinyal Input dan Sinyal Hasil Enkripsi



Gambar 11. Sinyal Input dan Sinyal Hasil Dekripsi

5. KESIMPULAN

Metode kriptografi RC4 berhasil diimplementasikan pada DSK TMS320C5402 sebagai metode yang digunakan untuk mengacak suara. Sistem tersebut bisa berhasil karena waktu eksekusi proses enkripsi/dekripsi tiap sampel tidak melebihi 125 μ detik sehingga sistem yang dihasilkan bisa real time. Dari hasil pengukuran terlihat waktu eksekusi yang dibutuhkan untuk proses enkripsi adalah 1,260 μ detik sedangkan waktu eksekusi yang dibutuhkan untuk proses dekripsi adalah 1,140 μ detik.

PUSTAKA

[1] R.L. Rivest, A. Shamir and L. Adelman, "On digital signatures and public keycryptosystems", *Commun.of the ACM*, Vol.21, No.2, pp.120-126, 1978.

[2] B. Schneier, "*Applied Cryptography: Protocols, Algorithms and Source Code in C* ", 2nd edition, John Wiley & Sons, Inc., USA, 2001

[3] A. J. Menezes, P.C.V. Oorschot and S.A. Vanstone, "*Handbook of Applied Cryptography*", CRC Press, 1996.

[4] Berkeley, "Choosing a DSP Processor. California: Berkeley Design Technology, Inc., http://www.BDTI.com.

[5] Texas Instruments (2000). SPRS079D: TMS320 VC5402 Fixed-Point Digital Signal Processor datasheet. USA: Texas Instruments.