Perancangan Perangkat Lunak Pengamanan Data Dengan Metode XTEA

Yudi[1], Marini[2]
Sistem Informasi, STMIK IBBI
Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548
E-mail: ynn_linc@yahoo.com

Abstrak

Kerahasiaan data adalah merupakan hal yang sangat penting dijaga dengan baik. Dari data dapat diperoleh informasi yang berguna. Tidak semua pihak berhak untuk mengetahui suatu informasi karena bersifat rahasia. Untuk mengatasi masalah dalam menjaga keamanan data maka dikembangkan teknik pengamanannya dengan teknik yang dikenal dengan kriptografi XTEA (TEA Extension) yang merupakan perbaikan atas algoritma TEA. XTEA salah satu algoritma kriptografi yang paling cepat dan efisien yang pernah ada. Mudah dan sederhana, masih terjamin keamanannya dan sangat cepat prosesnya. Hasil dari penelitian ini adalah sebuah program yang mampu mengenkripsi dan mendekripsi file serta string dengan menampilkan secara per langkah proses dari algoritma TEA dan XTEA serta menampilkan waktu eksekusi enkripsi dan dekripsi suatu file untuk mengetahui kecepatan dari algoritma.

Kata kunci: *Kriptografi, XTEA, Algoritma, Enkripsi, Dekripsi*

Abstract

Confidentiality of data is a very important thing maintained properly. From the data obtained can be useful information. Not all parties are entitled to know the information confidential. To overcome the problem of data security in safeguarding the security technique is developed with a technique known as cryptographic XTEA (TEA Extension) which is an improvement over the TEA algorithm. Cryptographic algorithm XTEA one of the fastest and efficient that ever existed. Easy and simple, it is still safe to consume and are very quick process. Results from this study is a program that is able to encrypt and decrypt files and strings to display the per-step process of TEA and XTEA algorithm and displays the execution time encryption and decryption of a file to determine the speed of the algorithm.

Keywords*: Cryptography, XTEA, Algorithm, Encryption, Decryption*

1. Pendahuluan

Data atau informasi merupakan salah satu elemen yang memegang peranan yang sangat besar dalam berbagai bidang kehidupan. Dengan semakin pesatnya perkembangan teknologi komputer, semakin banyak orang yang sanggup mengutak-atik data yang disimpan dengan rapi dan dijaga kerahasiaannya. Untuk mencegah terjadinya pengaksesan data oleh orang yang tidak berhak, maka dikembangkanlah berbagai teknik pengamanan data. Data yang tersimpan dalam arsip data itu dapat menyangkut kerahasiaan baik perseorangan maupun kelompok, jadi apabila komputer dan perlengkapan komunikasi dilengkapi dengan sistem pengamanan yang baik, maka komputer dan peralatan tersebut dapat menambah kepercayaan untuk menyimpan, mengolah, dan bertukar data atau informasi yang sangat penting [3]. Teknik pengamanan data yang biasanya dikenal sebagai kriptografi merupakan salah satu solusi untuk permasalahan di atas. Ada banyak metode kriptografi yang dapat digunakan untuk melakukan enkripsi data, antara lain: DES (*Data Encryption Standard*) yang dikembangkan oleh IBM, RSA (*Rivest-Shamir-Adleman Algorithm*), *Blowfish*, *Helix*, *Solitaire*, *RC4*, *Frog*, *GOST*, *SkipJack*, *TEA*, dan lain-lain. Metode-metode tersebut mempunyai tingkat keamanan yang bervariasi, kecepatan, dan kemudahan dalam hal implementasi algoritmanya. Algoritma TEA memiliki keunikan karena menggunakan algoritma yang efisien dengan menggunakan konsep Rasio Emas [4]. Di samping itu algoritma ini merupakan block cipher yang sederhana dan sangat cepat. Tetapi karena algoritma ini terdapat beberapa kelemahan terutama pada pembentukan kuncinya, oleh penciptanya algoritma ini direvisi dan dihasilkan algoritma baru yang diberi XTEA (TEA Extension). Dengan alasan tersebut maka dirancanglah suatu program yang mampu mengamankan *file* atau data. Secara umum tujuan penelitian adalah merancang perangkat lunak untuk pengamanan data dengan metode XTEA untuk menjaga kerahasiaan *file* atau data sehingga data tersebut tidak dapat diakses oleh pihak ketiga.

2. Metode Penelitian

2.1 Algoritma

Algoritma adalah program *kriptografi* yang digunakan untuk melakukan *enkripsi*. Ia bukanlah suatu kunci, tetapi menghasilkan kunci (dalam perancangan ini adalah "*password*")[1]. Suatu *algoritma*

yang kuat atau bagus akan menghasilkan *kriptografi* yang kuat dan bagus juga [5]. Dalam pembuatan perangkat lunak ini peneliti menggunakan *metode XTEA* atau disebut juga dengan *TEA Extension*,

2.2 Metode XTEA

XTEA, atau disebut juga dengan *TEA* Ext*e*nsion, yaitu perbaikan atas *algoritma* TEA. *XTEA* salah satu *algoritma kriptografi* yang paling cepat dan efisien yang pernah ada [2]. Mudah dan sederhana, masih terjamin keamanannya dan sangat cepat prosesnya. Dikembangkan oleh *David Wheeler* dan *Roger Needham*. Merupakan kriptografi *symmetric block cipher* karena menggunakan kunci rahasia dan dioperasikan pada 64 bit pesan sekaligus dimana kunci rahasia dengan panjang 128 bit dibagi antara pemakai. Enkripsi dilakukan dengan operasi yang tidak bersifat merusak data seperti *SHIFT*, *XOR*, *DEL*, *AND* dan ADD. Deskripsi hanya bisa dilakukan apabila kunci rahasia diketahui.

3. Analisa & Hasil

3.1 Analisa Sistem

Ada dua hal yang penting yang perlu dalam melindungi folder ini yaitu : keamanan dan kecepatan. Untuk tujuan keamanan, enkripsi/redirect tidak hanya dilakukan pada isi *file* saja, melainkan juga struktur *folder*. Jika seseorang yang tidak berhak mengetahui isi suatu *folder* yang terkunci dapat menemukan struktur *folder* tersebut, ia bisa saja melakukan analisis terhadap *file* terenkripsi yang ia temukan. Jika ia dapat menemukan cara untuk mendekripsi satu *file* tersebut, ia dapat lebih mudah menemukan cara untuk mendekripsi keseluruhan *file* dalam *folder* tersebut. Untuk melakukan penyembunyian struktur *folder*, seluruh *file* dalam *folder* yang dikunci disatukan dalam satu *file* khusus. *File* khusus tersebut mengandung kumpulan *file* yang terenkripsi beserta pengaturan *file* tersebut dalam *folder* yang dikunci. Kecepatan, pengguna tentu menginginkan respon yang cepat dari suatu aplikasi yang dijalankan. Ia tidak ingin menunggu proses lain dari aplikasi tersebut yang tidak diperlukannya. Misalnya jika ia ingin mendekripsi satu *file* saja, ia tidak perlu mendekripsi semua *file* yang terdapat pada *folder* yang terproteksi karena ia tidak memerlukan *file-file* lain dan tidak ingin menunggu aplikasi memproses pendekripsian keseluruhan isi *folder*. Untuk mendapatkan *file* atau *folder* yang berada dalam *folder* yang diproteksi, aplikasi cukup mengakses posisi tertentu pada *file* khusus tersebut. Seperti yang telah disebutkan di atas, *file* khusus tersebut mengandung pengaturan *file-file* yang diproteksi. Pengaturan tersebut berupa struktur *tree* yang merepresentasikan *folder*.

3.2 HASIL PERANCANGAN

3.2.1 Perancangan Sistem Perangkat Lunak

3.2.1.1 Perancangan *Form*

Perancangan perangkat lunak untuk melindungi folder (folder proteksi) dirancang dengan menggunakan visual basic 6.0. Program ini hanya dibuat dalam dua tampilan *form* yaitu *form* utama dan *form about*. Tampilan *form* utama ditunjukkan pada Gambar 3.1 berikut ini.

Gambar 1. Rancangan *form* utama

Pada *form* utama ini digunakan komponen SSTab yang berguna untuk membagi proses pada program ini menjadi bagian: teks yang berisi keterangan tentang TEA dan XTEA pada bagian “Pengenalan TEA dan XTEA”, bagian untuk memproses *file* adalah pada bagian SSTab “Proses File” dan bagian untuk memproses *string* pada bagian SSTab “Proses String”.

Pada *form* ini juga terdapat dua buah *command button* yaitu “Tentang” untuk memunculkan *form about*. Kedua adalah *command button* “Keluar” yang berfungsi untuk keluar dari program ini.

Untuk menampilkan status proses dan keterangan *file* yang diproses maka pada bagian bawah *form* diletakkan objek *status bar*. Keterangan yang dapat dilihat oleh user seperti: status proses, nama *file* yang diproses, ukuran *file* sebelum dan sesudah diproses, serta keterangan kecepatan proses dalam satuan Kbyte / detik.

Seperti disebutkan di atas objek SSTab terdapat tiga buah Tab seperti terlihat pada Gambar 2. sampai Gambar 3.

Gambar 2. Rancangan pengenalan TEA dan XTEA

Pada rancangan *Tab* 1 terdapat keterangan dari algoritma TEA dan XTEA sedangkan dua buah *picture box* disisi kanan dan kiri memuat diagram siklus *round* dari TEA dan XTEA. *Picture box* pada bagian tengah menampilkan tabel perbedaan antara TEA dan XTEA.

Gambar 3. Rancangan proses *file*

Tab 2 digunakan untuk memproses *file*. Pada program ini semua jenis *file* dapat diproses. *File List* berfungsi untuk menampilkan semua *file* yang terdapat pada *folder* aktif *Dir List Box*.

Tab berikutnya adalah *Tab* 3 yang dipakai untuk memproses *string*, ada dua bentuk tampilan pada *Tab* ini yaitu bentuk tampilan saat proses enkripsi dan dekripsi. Perbedaannya hanya pada teks *Label* yang berubah menjadi teks "Enkripsi" jika *option button* "Enkripsi" diklik dan sebaliknya. Sisi kanan dari tab ini merupakan tampilan proses TEA dan tampilan di sisi kanan untuk proses XTEA, tujuannya agar *user* dapat dengan mudah membandingkan cara kerja kedua algoritma. Bagian tengah merupakan bagian *input* berupa teks, *password* atau kunci, jumlah *round*, dan penentuan proses enkripsi atau dekripsi, juga terdapat tampilan *flow chart* dari kedua algoritma.

Hasil perhitungan dan proses enkripsi dan dekripsi kedua algoritma ditampilkan pada bagian Text Box di bagian tengah sisi kiri dan kanan. Tampilan algoritma keduanya berada pada bagian atas. Hasil proses akan ditampilkan di bagian sisi kiri dan kanan bawah berupa teks dan dalam bentuk heksadesimal. Berikut pada gambar 4. dan 5. akan dijelaskan tentang Rancangan Proses Enkripsi *String* dan Rancangan Proses Dekripsi *String*

Gambar 4. Rancangan proses enkripsi *string*

Form About ini berfungsi untuk menampilkan nama pembuat program beserta keterangan program. *Form* ini dapat diakses melalui *form* utama dengan menekan tombol "Tentang". Objek utama untuk merancang *form* ini hanya berupa *Label* untuk menampilkan teks dan *picture box* untuk menampilkan sebuah gambar.

Gambar 5 Rancangan proses dekripsi *string*

3.2.1.2 Perancangan *Class Module*

Perancangan *Class Module* bertujuan agar objek *class* yaitu berisi fungsi-fungsi utama proses enkripsi / dekripsi dengan algoritma dapat dipakai dengan mudah dan dapat dipakai kembali (*reuseable*) untuk pembuatan program lain yang memakai algoritma enkripsi XTEA juga. Selain itu fungsi yang dideklarasikan berupa objek *class* akan lebih cepat dalam hal pemrosesan.

Class Module yang dibuat diberi nama clsXTEA (singkatan dari Class XTEA). *Class* ini berisi rutin-rutin dari algoritma XTEA seperti fungsi untuk enkripsi *file*, dekripsi *file*, operasi XOR, ADD, DEL, *Shift Left*, *Shift Right* dan AND, serta suatu fungsi untuk mengecek keberadaan suatu *file* pada lokasi *folder* tertentu.

3.2.1.3 Perancangan *Module Function*

Module Function berguna untuk mendeklarasikan semua fungsi yang berhubungan dengan operasi pada *file*, mengambil *path* pada *file*, mengambil nama *file*, ekstensi *file*, mengecek tanggal pembuatan *file*, ukuran *file*, jenis atribut *file*, dan lain-lain. Kebanyakan fungsi yang dideklarasikan tersebut merupakan fungsi yang dikompilasi dalam *library* atau pustaka pada sistem operasi Windows. Jadi fungsi tersebut sebenarnya tidak dibuat lagi sendiri oleh penulis melainkan langsung menggunakannya melalui *Visual Basic*.

3.3 Implementasi Program

Implementasi sistem program ini mencakup spesifikasi kebutuhan perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*).

3.3.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Program ini dijalankan dengan menggunakan perangkat keras (*hardware*) yang mempunyai spesifikasi minimal adalah sebagai berikut : *prosesor Intel Pentium* IV 2.0 GHz, m*emory* 256 MB, *harddisk* 250 GB, VGA *card* 256 MB, monitor dengan resolusi 1024 × 768 *pixel*, m*ouse, Keyboard.*

Adapun perangkat lunak (*software*) yang digunakan untuk menjalankan aplikasi ini adalah lingkungan sistem operasi *MS-Windows 98* atau *MS-Windows NT/2000/XP.*

3.3.2 Cara Instalasi

Program ini tidak memerlukan cara instalasi yang khusus, dengan alasan bahwa semua *file* yang dibutuhkan oleh aplikasi ini dapat dikompilasi menjadi satu *file executeable*. Jadi untuk instalasi program ini cukup dengan meng-*copy file executeable*-nya (XTEA.EXE) ke dalam lokasi *folder* yang dipilih pada *harddisk*. Jika program tidak dapat dijalankan lakukanlah instalasi dengan menjalankan *file* SETUP.EXE.

3.3.3 cara penggunaan program

Untuk menggunakan program ini, jalankan *file executeable* (XTEA.EXE) dari lokasi di mana *file* tersebut di-*copy*-kan, ataupun jika program diinstalasi maka *icon shortcut* program dapat dijalankan dari tombol *Start → Program Windows*. Setelah itu jika program dijalankan maka akan tampak tampilan seperti gambar berikut ini.

Gambar 6 Tampilan Awal Program

Pada tampilan di atas merupakan tampilan tab pertama yang berisi keterangan mengenai TEA dan XTEA. Bagian berikutnya adalah tab kedua untuk melakukan proses enkripsi atau dekripsi. Untuk memilih *file* yang akan diproses maka terlebih dahulu *user* dapat memilih *drive* yang terdapat pada komputer dan *folder* pada bagian sisi kiri. Setelah *folder* dipilih maka otomatis semua *file* yang terdapat pada *folder* tersebut akan ditampilkan pada *File List*

.

Gambar 7. Tampilan Proses *File*

Langkah selanjutnya adalah menentukan *folder* tujuan dimana hasil *file* yang diproses ini akan disimpan dengan mengklik pada *command button* bertanda "…" maka akan dimunculkan kotak dialog "*Browse for Folder*" untuk memilih *folder* tempat *file* disimpan. *Path* yang dipilih tersebut akan ditampilkan pada *Combo box folder* tujuan dimana pada *Combo box* tersebut setiap *path* yang diinput akan tetap berada dalam *list* sehingga memudahkan *user* untuk memilih *folder* terdahulu.

Berikutnya adalah memasukkan kata kunci (*Password*). Jumlah karakter maksimum adalah 16 karakter (128 bit). Pilihan *check box* "Sembunyikan *Password*" jika ditandai maka saat input *password* akan bertanda "*". Untuk algoritma XTEA ini *user* dapat menentukan jumlah *round* dimulai dari 1 hingga 64 *round*. Jika *user* mengosongkan pilihan ini maka secara default *round* akan bernilai 32 saat diproses. Dua *option button* "Enkripsi" dan "Dekripsi" digunakan untuk menentukan proses.

Untuk proses maka lakukan klik ganda pada *file* yang diinginkan maka akan dimunculkan sebuah kotak dialog konfirmasi apakah *file* akan dienkripsi atau didekripsi. Setelah itu *file* akan diproses, kemajuan proses dapat dilihat dari *progress bar.*

Gambar 8. Tampilan Proses Enkripsi *File*

Bagian berikutnya adalah tab ketiga untuk proses *string*. Pada bagian ini sisi kiri untuk perhitungan algoritma TEA dan sisi kanan untuk perhitungan algoritma XTEA. Untuk melakukan proses enkripsi atau dekripsi pada string dapat ditentukan dengan mengklik pada *option button* "Enkripsi" dan "Dekripsi" kemudian menekan tombol "Proses".

Gambar 9. Tampilan Proses *String*

Maka proses akan dilakukan dan alur proses dapat dilihat pada kedua *flow chart* pada bagian bawah. Urutan keseluruhan proses dapat dilihat pada bagian *text box* "Rangkaian Proses…" dan hasil per-

hitungan nilai dapat ditentukan pada bagian “Ringkasan Proses…”. Hasil *cipher* teks atau *plain* teks akan ditampilkan di bagian bawah dalam bentuk ASCII dan heksadesimal.

Gambar 10. Tampilan Hasil Enkripsi *String*

3.3.4 Pengujian

Untuk mengetahui hasil pengujian program ini dengan algoritma TEA yang telah diimplementasikan maka dilakukan pengujian pada beberapa jenis *file* seperti *file* teks, *audio*, *graphic*, dan *file word processsor* dengan menggunakan kunci (*password*) “*XTEA*”. Pengujian dilakukan dengan menggunakan spesifikasi komputer yang seperti berikut ini: *Processor Intel Pentium IV 2.0 GHz dengan L2 Cache 512 KB, Motherboard Asus P4S533-E Deluxe Socket 478 Chipset Intel 845E, VGA Card Asus V7100 Pro 64 MB GeForce2 MX-400, Memori 4 × 184-pin DIMM Sockets 512 MB ECC* dan *Harddisk ATA100 Seagate Barracuda 7200 rpm 120 GB & Sistem Operasi Windows XP.*

Tabel 1. Tabel Hasil Pengujian Proses Enkripsi

Jenis *File*	**Ukuran *File* (byte)**	**Ukuran *File Output* (byte)**	**Lama Proses (Kbyte/detik)**			
			Test I	**Test II**	**Test III**	**Rata-Rata**
Gambar	5.385	5.400	25,98	22,44	25,98	24,80
Gambar	6.818	6.832	28,39	26,69	26,69	27,25
Text	311	328	5,17	5,17	5,08	5,14
Text	899	912	6,36	8,17	8,10	7,54
Audio	21.312	21.328	40,36	38,08	39,22	39,22
Audio	668.440	668.456	50,77	50,89	50,64	50,76
EXE	290.816	290.832	54,58	54,91	53,47	54,32
EXE	1.176.379	1.176.392	50,95	51,09	49,58	50,54

Tabel 2. Tabel Hasil Pengujian Proses Dekripsi

Jenis *File*	**Ukuran *File* (byte)**	**Ukuran *File Output* (byte)**	**Lama Proses (Kbyte/detik)**			
			Test I	**Test II**	**Test III**	**Rata-Rata**
Gambar	5.400	5.385	12,03	12,49	12,95	12,49
Gambar	6.832	6.818	13,32	13,32	12,90	13,18
Text	328	311	3,89	3,84	4,90	4,21
Text	912	899	4,69	5,59	5,13	5,14
Audio	21.328	21.312	15,31	16,06	15,86	15,74
Audio	668.456	668.440	17,45	16,52	17,37	17,11
EXE	290.832	290.816	17,38	17,38	17,39	17,38
EXE	1.176.392	1.176.379	17,46	17,50	17,48	17,48

Dari hasil yang diperoleh menunjukkan bahwa proses dekripsi lebih cepat dibandingkan dengan proses enkripsi.

4. Kesimpulan

Perangkat lunak metode *XTEA* dirancang secara sangat menarik dari segi tampilan dan juga efektif penggunaanya. Perangkat lunak ini sangat membantu pengguna untuk menyimpan data-data rahasia yang tidak ingin orang lain dapat membukanya. Pada algoritma *XTEA* kunci telah dicampurkan dan dioperasikan bersamaan dengan *plain* teks. Jika seseorang mengirimkan *cipher* teks kepada orang lain maka ia harus mengirim kunci dan jumlah putaran yang sama kepada penerima agar proses *dekripsi* dapat berhasil. Dari hasil yang diperoleh menunjukkan bahwa proses dekripsi lebih cepat dibandingkan dengan proses *enkripsi*. Pengembangan algoritma XTEA dari TEA membuat algoritma ini lebih aman tetapi waktu proses menjadi lebih lama.

Buku Teks :

[1] Ariyus D. 2006. *Kriptografi Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.
[2] Arryawan E, Smitdev Community. 2010. *Password is Nothing*. Gramedia. Jakarta.
[3] Collberg C., Nagra J. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*, Upper Saddle River, NJ: Addison-Wesley, 2010.
[4] Patterson, Wayne. *Mathematical Cryptology for Computer Scientiests and Mathematicians*. The United States of America : Rowman & Littlefield Publishers. 1987.
[5] Scheiner B. *Applied Cryptography.* 2nd. John Wiley & Sons. 1996.