

Fungsi Hash SNEFRU dan Metode Enkripsi IDEA untuk Keamanan Dokumen Elektronik

Gregorius S. Budhi

Dosen Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
Email: greg@petra.ac.id

Justinus Andjarwirawan

Staff Pusat Komputer Universitas Kristen Petra

Radius Indrawan

Alumni fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

ABSTRAK

Dokumen-dokumen penting yang hanya boleh diketahui dan digunakan oleh pihak tertentu, memerlukan sistem pengamanan dokumen yang mampu mencegah serangan dari pihak lain. Dengan menggabungkan metode enkripsi IDEA (International Data Encryption Algorithm) untuk mengenkripsi isi dokumen dan fungsi hash SNEFRU untuk mengetahui adanya perubahan terhadap isi dokumen diharapkan tingkat keamanan dokumen menjadi lebih baik. Hasil penerapan ini berupa sebuah perangkat lunak yang mampu melakukan enkripsi dan dekripsi terhadap semua bentuk dokumen, meng-generate security key unik berdasarkan dokumen yang dienkripsi, serta memberikan laporan yang akurat mengenai ada atau tidaknya perubahan terhadap isi dokumen. Pengujian terhadap perangkat lunak dibagi menjadi dua tahap, yaitu pengujian sistem dalam menjalankan proses enkripsi dan dekripsi serta pengujian kecepatan sistem. Dari hasil pengujian diketahui bahwa perangkat lunak mampu menjalankan proses enkripsi dan dekripsi dengan baik, serta dapat mengetahui bila terjadi modifikasi terhadap isi dokumen.

Kata kunci: Metode Enkripsi IDEA, Fungsi Hash SNEFRU, Keamanan Dokumen.

1. PENDAHULUAN

Saat ini proses pertukaran dan *sharing* dokumen lebih sering dilakukan, sehingga dibutuhkan pula suatu sistem keamanan yang baik untuk mencegah serangan-serangan dari pihak lain, khususnya terhadap dokumen-dokumen penting yang hanya boleh diketahui dan digunakan pihak-pihak tertentu saja.

International Data Encryption Algorithm (IDEA) adalah sebuah metode yang menggunakan *key* 128-bit sehingga dokumen yang dienkripsi cukup aman. Fungsi *hash* SNEFRU dipakai untuk mengetahui apakah isi suatu dokumen telah diubah atau tidak. Masukan untuk fungsi ini sepanjang 512-bit, sedangkan keluarannya merupakan angka sepanjang 128-bit atau 256-bit.

Dengan penggunaan metode IDEA dan SNEFRU secara bersamaan untuk membangun sebuah sistem keamanan dokumen, diharapkan tingkat *Confidentiality* dan *Integrity* dokumen akan lebih baik.

2. TEORIPENUNJANG

2.1 *International Data Encryption Algorithm* (IDEA)

IDEA adalah algoritma enkripsi yang dikembangkan oleh Xuejia Lai dan James Massey, dan diharapkan mampu menggantikan DES. IDEA menggunakan

sebuah *key* dengan panjang 128-bit untuk mengenkripsi data dalam blok-blok sepanjang 64-bit[1,5].

Beberapa karakteristik IDEA antara lain[1,5]:

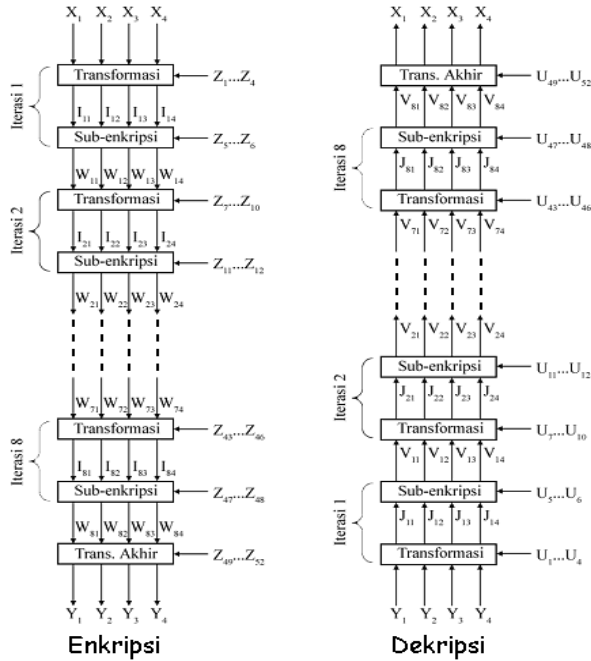
- Blok dengan panjang 64-bit dianggap cukup kuat untuk menghindari analisa statistik terhadap pola-pola blok.
- *Key* dengan panjang 128-bit memungkinkan pencegahan terhadap tindakan pencarian *key*.
- Menggunakan tiga operasi berbeda di mana setiap operasi membutuhkan dua masukan sepanjang 16-bit untuk menghasilkan satu keluaran sepanjang 16-bit.
- Tiga operasi tersebut adalah:
 - Bit-per-bit *exclusive-OR*.
 - Penjumlahan bilangan bulat *modulo* 2^{16} (65536), dengan lambang \boxplus .
 - Perkalian bilangan bulat *modulo* $2^{16}+1$ (65537), dengan lambang: \odot . Sebuah blok yang seluruh bitnya terdiri dari nol dianggap sebagai 2^{16} .

Secara garis besar algoritma IDEA dapat digambarkan seperti pada gambar 1.

52 blok *subkey* untuk enkripsi ($Z_1 - Z_{52}$) dan dekripsi ($U_1 - U_{52}$), dengan panjang masing-masing 16-bit, didapat dari 128-bit *key* enkripsi.

Skema proses *subkey* enkripsi ($Z_1 - Z_{52}$) adalah sebagai berikut: 8 blok *subkey* pertama, diberi label Z_1, Z_2, \dots, Z_8 , diambil langsung dari *key*, dimana Z_1

adalah 16-bit paling kiri dari *key*. Setelah itu dilakukan *circular left-shift* sebanyak 25-bit, kemudian Z_9 hingga Z_{16} didapat dengan cara yang sama dengan Z_1 hingga Z_8 . Langkah ini diulang terus-menerus hingga didapat 52 blok *subkey* [1,5].



Gambar 1. Enkripsi dan Dekripsi IDEA untuk tiap 64-bit dokumen [1]

Subkey untuk dekripsi ($U_1 - U_{52}$) didapat dari *subkey* untuk enkripsi dengan sedikit perubahan sebagai berikut [1,6]:

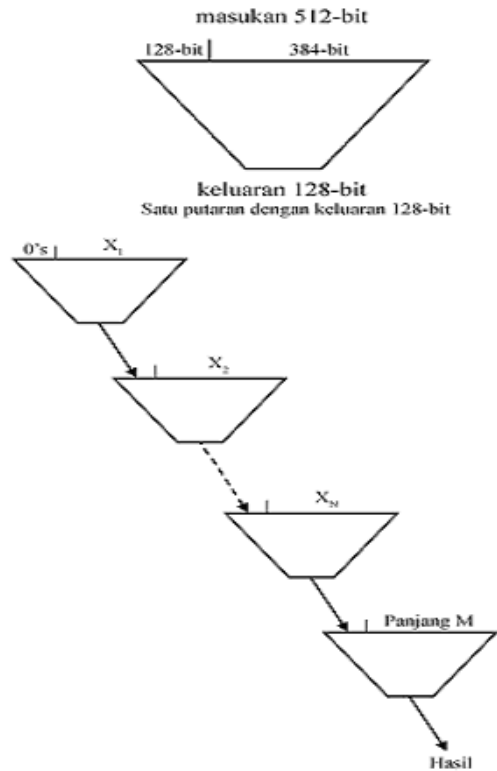
1. Untuk *subkey* U dengan index $i \text{ mod } 6 = 1$
 $U_i = Z_j^{-1}$ dimana $Z_j \odot Z_j^{-1} = 1$
2. Untuk *subkey* U dengan index $i \text{ mod } 6 = 2$ atau $i \text{ mod } 6 = 3$
 $U_i = -Z_j$ dimana $-Z_j \boxplus Z_j = 0$
3. Untuk *subkey* U dengan index $i \text{ mod } 6 = 5$ atau $i \text{ mod } 6 = 0$
 $U_i = Z_j$

2.2 SNEFRU

SNEFRU dikembangkan oleh Ralph Merkle dengan tujuan untuk memberi fingerprint pada sebuah dokumen sehingga keaslian dokumen tersebut terjaga dan memberi kemudahan implementasi pada prosesor 32-bit.

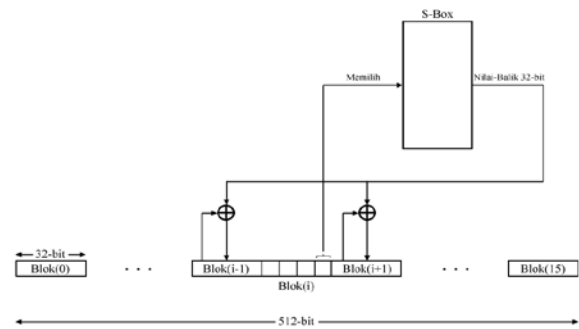
SNEFRU adalah sebuah fungsi *hash*, H , yang menghasilkan sebuah kode *hash* untuk pesan dengan panjang berapa saja. Fungsi H menggunakan iterasi dari fungsi yang lebih sederhana, H_{512} , yang memetakan sebuah masukan dengan panjang 512-bit menjadi keluaran dengan panjang k -bit. Untuk tiap

iterasi, sebagai masukan, H_{512} mengambil keluaran dari iterasi sebelumnya (dengan panjang k -bit) ditambah dengan satu blok dari pesan (512-k bit) [1].



Gambar 2. Fungsi Hash SNEFRU

Untuk sistem yang dirancang, keluaran proses SNEFRU yang digunakan adalah 128 bit. Sedangkan proses didalam setiap blok 512-bit dapat dilihat pada gambar 3.



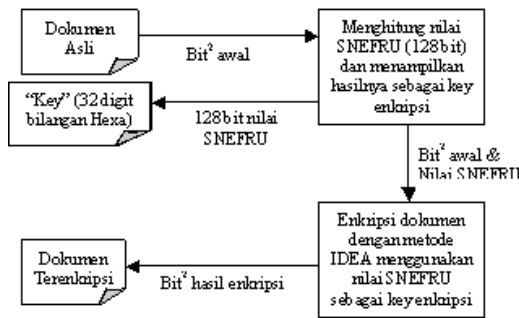
Gambar 3. Proses pada setiap blok 512-bit

Diambil dari: Stallings, William, *Network and Internetwork Security Principles and Practice*. New Jersey: Prentice Hall, Inc., 1995, page 184.

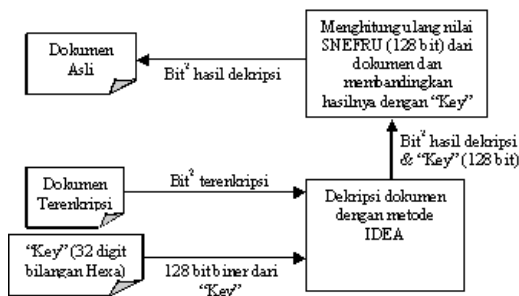
Setelah semua operasi XOR dijalankan, dilakukan *circular right-shift* untuk masing-masing *word* 32-bit. Tiap *word* dirotasi satu atau lebih oktet (kelipatan 8-bit) tergantung pada jadwal 16,8,16,24. Terakhir, setelah semua *pass* selesai, *word* pada blok dipasangkan silang: 0 dengan 15, 1 dengan 14, dan seterusnya [1].

3. DESAIN SISTEM

Desain Sistem secara keseluruhan dapat dilihat pada blok diagram – blok diagram berikut ini:



Gambar 4. Blok Diagram Proses Enkripsi



Gambar 5. Blok Diagram Proses Dekripsi

3.1 Proses SNEFRU

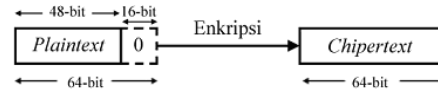
Berdasarkan teori, proses pembacaan terhadap dokumen tergantung pada panjang keluaran proses SNEFRU. Dalam sistem ini digunakan keluaran SNEFRU sepanjang 128-bit sehingga pembacaan terhadap dokumen dilakukan tiap 384-bit.

Untuk dokumen yang memiliki panjang kurang dari 384-bit perlu dilakukan penyesuaian terlebih dahulu dengan cara membulatkan panjang hasil pembacaan terhadap isi dokumen sehingga mencapai 384-bit. Pada sistem ini, pembulatan dilakukan dengan cara menambahkan bit 0 di belakang hasil pembacaan hingga panjangnya mencapai 384-bit. ‘Pass’ (pengulangan proses) dilakukan sebanyak 4 kali, sesuai saran dari Merkle[1].

3.2 Proses IDEA

Untuk melakukan enkripsi IDEA dibutuhkan masukan berupa *plaintext* dengan panjang minimal 64-bit untuk diproses menjadi *chipertext* dengan panjang 64-bit pula. Dokumen yang ukurannya lebih besar dari 64-bit, maka proses pembacaan dilakukan secara sekuensial, per 64-bit dari bit awal dokumen tersebut. Untuk dokumen / bagian akhir dari dokumen yang ukurannya lebih kecil dari 64-bit, harus dibulatkan menjadi 64-bit

terlebih dahulu sebelum dilakukan proses enkripsi. Proses pembulatannya dilakukan dengan cara menambakan satu atau lebih bit 0 di belakang dokumen sehingga panjangnya menjadi 64-bit, seperti terlihat pada contoh gambar 6.



Gambar 6. Contoh Proses Enkripsi Untuk Dokumen 48-bit

3.2.1 Header IDEA

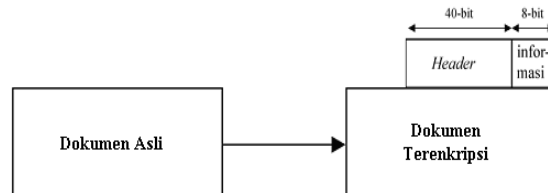
Proses enkripsi dapat dijalankan terhadap dokumen apa saja, termasuk dokumen yang telah dienkripsi. Lain halnya dengan proses dekripsi, dekripsi hanya dapat dilakukan terhadap dokumen-dokumen yang telah dienkripsi saja, karena proses dekripsi digunakan untuk mengembalikan hasil enkripsi menjadi seperti semula. Oleh karena itu diperlukan ‘header’ yang ditambahkan pada dokumen yang telah dienkripsi.

Header tersebut berfungsi sebagai tanda bahwa dokumen tersebut adalah hasil enkripsi. Apabila ditemukan header pada dokumen, dapat dipastikan bahwa proses dekripsi dapat dilakukan pada dokumen. Lokasi penempatan header harus konsisten sehingga memudahkan proses pencarian. Pada sistem ini header yang digunakan adalah kata ‘IDEA’ sepanjang 40 bit (gambar 7).

3.2.2 Informasi Tambahan IDEA

Seperti yang telah dijelaskan sebelumnya, untuk proses enkripsi terhadap dokumen-dokumen yang ukurannya bukan kelipatan 64-bit dibutuhkan suatu proses pembulatan. Hal ini mengakibatkan adanya perbedaan isi antara dokumen asli dengan dokumen hasil pembulatan. Kesulitannya adalah untuk mendeteksi berapa jumlah bit 0 yang ditambahkan diakhir dokumen yang telah dienkripsi.

Guna mengatasi kesulitan tersebut perlu ditambahkan informasi pada dokumen yang telah dienkripsi, tentang jumlah bit dari blok terakhir dokumen yang telah dibagi per 64-bit. Informasi disimpan dalam satuan *byte* (8-bit), yaitu bilangan antara 1 hingga 64, dan diletakkan setelah header IDEA (gambar 7).



Gambar 7. Peletakan Header dan Informasi Tambahan

3.2.3 Proses Enkripsi Berulang Terhadap Dokumen

Karena proses enkripsi dirancang untuk berjalan pada segala bentuk dokumen, termasuk dokumen yang telah dienkripsi, maka sistem dapat pula melakukan enkripsi berulang terhadap dokumen dan menghasilkan *key* – *key* yang berbeda.

Untuk mengembalikan dokumen hasil enkripsi berulang ini dapat dilakukan proses dekripsi berulang dengan memperhatikan sequent *key* yang digunakan untuk tiap prosesnya. *Key* yang dihasilkan dari proses enkripsi paling akhir menjadi *key* untuk proses dekripsi paling awal.

4. ANALISA DAN PENGUJIAN SISTEM

4.1 Analisa Sistem

4.1.1 Pemilihan Metode

Dengan menerapkan algoritma enkripsi IDEA pada sebuah dokumen, maka tingkat *Confidentiality* dokumen dari serangan “Intruder Luar” bertambah. Pemilihan metode IDEA ini didasari atas beberapa alasan sebagai berikut:

- Algoritma enkripsi yang dikembangkan oleh Xuejia Lai dan James Massey ini tergolong baru, dikembangkan pada tahun 1990[1,5].
- Dewasa ini algoritma enkripsi IDEA banyak digunakan untuk sebagai metode enkripsi di banyak sistem dan bebas (free) untuk digunakan pada sistem Non-Komersial[8].
- Dengan menggunakan 128 – bit key yang sulit dihandel oleh komputer tradisional (Stand-Alone PC) dan memerlukan kurang lebih $3E+38$ kombinasi untuk memecahkannya[1].
- Proses didalam IDEA yang menggunakan perhitungan matematis bersifat lebih ‘secure’ bila dibandingkan dengan metode - metode enkripsi lain yang menggunakan tabel konversi tertentu (DES / 3DES)[1].
- Sampai sekarang peneliti belum mendengar berita sukses tentang usaha pemecahan “Cipher Document” hasil dari algoritma IDEA.

Penggunaan fungsi Hash SNEFRU pada sistem dapat meningkatkan *Integrity* dokumen dari serangan “Intruder Dalam”. Adapun alasan penggunaan SNEFRU, adalah sbb:

- Fungsi Hash yang dikembangkan oleh Ralph Merkle ini tergolong baru, dikembangkan pada tahun 1990[1,2,3].
- Kelemahan dari SNEFRU 2 – pass, yaitu, value yang sama untuk karakter yang berbeda, telah dibuktikan oleh Eli Biham dan Adi Shamir pada

tahun 1991, tapi untuk 3 – pass dan seterusnya belum terbukti sampai sekarang[1,4].

- Hasil SNEFRU 128 bit dapat dijadikan key untuk enkripsi IDEA.
- Proses didalam SNEFRU yang menggunakan perhitungan matematis dan tabel S-Box yang digenerate secara random dianggap cukup ‘secure’ oleh peneliti.

4.1.2 Perubahan Jumlah Byte Pada Dokumen yang Dienkripsi

Pada saat enkripsi, proses SNEFRU tidak menambah atau mengurangi jumlah byte dokumen asli. Proses ini untuk menggenerate *fingerprint* sepanjang 128-bit dari dokumen yang digunakan untuk ‘key’ enkripsi IDEA. Pada saat proses dekripsi, ‘key’ juga digunakan untuk memeriksa *Integrity* dokumen, dengan cara membandingkannya dengan *fingerprint* dari dokumen hasil dekripsi.

Sementara itu proses IDEA menghasilkan penambahan byte pada dokumen hasil enkripsi dalam jumlah hampir konstan, yaitu:

Total Tambahan Bytes = Pembulatan keatas kelipatan 8 terdekat + Ukuran header IDEA + Ukuran informasi tambahan IDEA = 0 s/d 7 bytes + 5 bytes + 1 byte = 6 s/d 13 bytes

4.2 Analisa Pengujian Empiris

Pengujian sistem yang dilakukan adalah pengujian kecepatan sistem terhadap besar file yg diproses, dengan peralatan sebagai berikut:

Stand Alone PC: Pentium III, RAM 128Mb

Compiler: Delphi 5

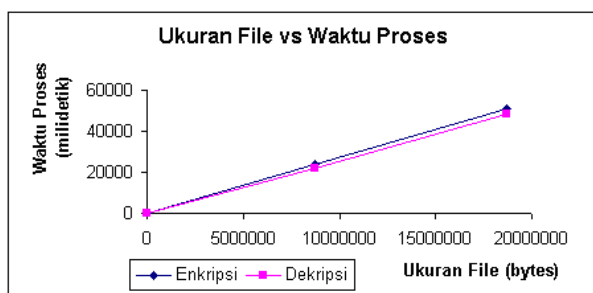
Sample Files:

1. A.txt
2. Madonna - Die Another Day.mp3
3. Win2KXP 44.03.zip

Hasil pengujian dapat dilihat pada tabel 1. dan Gambar 8.

Tabel 1. Data Hasil Pengujian Ukuran File terhadap Waktu Proses

Nama File	Enkripsi		Dekripsi	
	Ukuran (bytes)	Waktu Proses (milidetik)	Ukuran (bytes)	Waktu Proses (milidetik)
A.txt	68	60	72	60
Madonna - Die Another Day.mp3	8.726.528	23.654	8.726.534	22.172
Win2KXP 44.03.zip	18.675.920	50.914	18.675.924	48.109



Gambar 7. Grafik Hasil Pengujian Ukuran File terhadap Waktu Proses

Dari hasil pengujian didapat fakta bahwa:

- Proses enkripsi dan dekripsi untuk sebuah file yang sama, memiliki kecepatan proses kurang lebih sama.
- Waktu proses enkripsi dan dekripsi berbanding lurus dengan besar file yang diproses, dengan Running Time $T(n) = O(n)$ (Gambar 4.1).
- Waktu Proses masuk akal untuk diaplikasikan pada dunia nyata. Hal ini berdasarkan fakta bahwa untuk memproses file sebesar 18Mb dibutuhkan waktu selama kurang lebih 50 detik, pada processor Pentium III.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

- Sistem dapat digunakan untuk melakukan enkripsi dan dekripsi terhadap segala bentuk dokumen, termasuk dokumen yang telah dienkripsi sebelumnya.
- Penggunaan algoritma enkripsi IDEA dan fungsi hash SNEFRU secara bersamaan akan meningkatkan *Confidentiality* dan *Integrity* dokumen.
- Dari hasil pengujian kecepatan proses dapat disimpulkan bahwa sistem dapat diaplikasikan pada dunia nyata.

5.2 Saran

- Penggunaan metode *Public Key Cryptosystem*, seperti RSA atau DSA, pada saat pengiriman 'key' ke tujuan, dapat menjamin *Authentication* dokumen.
- Fungsi hash SNEFRU dapat diganti dengan fungsi hash lain yang lebih baru, misal MD4 atau MD5. Yang perlu diperhatikan adalah, fungsi hash pengganti itu harus mengeluarkan nilai hasil sepanjang 128-bit.

DAFTAR PUSTAKA

- [1] Stallings, William, *Network and Internetwork Security Principles and Practice*. New Jersey: Prentice Hall, Inc., 1995
- [2] Merkle, Ralph C, "Fast Software Encryption Functions", *Crypto '90*, pages 476-501
- [3] Merkle, Ralph C, "A Fast Software One-Way Hash Function", *Journal of Cryptology*, 3 (1990), pages 43-58
- [4] Biham, Eli, Shamir, Adi, "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer", *Crypto '91*, pages 156-171
- [5] Lai, X., Massey, J., "A Proposal for a New Block Encryption Standard", *Proceedings, EUROCRYPT '90*, 1990.
- [6] -----, "IDEA", <http://www.momentus.com.br/PGP/doc/idea.html>.
- [7] -----, "IDEA-Algorithm", <http://www.funet.fi/pub/crypt/cryptography/symmetric/idea/idea-algorithm.txt>.
- [8] -, "Cryptographic Algorithms", <http://www.eskimo.com/~weidai/algorithms.html#symmetric>.
- [9] -----, "Encryption, Digital Signatures, and Certification Authorities", [http:// dependability.cs.virginia.edu/bibliography/can-ch03.pdf](http://dependability.cs.virginia.edu/bibliography/can-ch03.pdf).
- [10] -----, <http://www.cs.uu.nl/wais/html/na-dir/cryptography-faq/part07.html>
- [11] -----, "Snefru", <ftp://arisia.xerox.com/pub/hash/hash2.5a/>