PIE: Block Cipher Algorithm

Dininta Annisa
Program Studi Teknik Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
dinintaannisa@gmail.com

Pipin Kurniawati
Program Studi Teknik Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
pipiiinn@gmail.com

***Abstrak*—Enkripsi pesan banyak digunakan untuk menjaga keamanan komunikasi. Sudah banyak algoritma enkripsi yang telah dikembangkan. Algoritma tersebut harus dirancang serumit mungkin agar sukar untuk dipecahkan. Makalah ini berisi rancangan dan analisis dari sebuah algoritma enkripsi *block cipher* baru bernama algoritma PIE. Algoritma ini menggunakan dua buah jaringan feistel yang mengandung fungsi substitusi dan transposisi di dalamnya. Algoritma ini dirancang sedemikian rupa agar memenuhi prinsip *confusion* dan *diffusion*.**

Kata kunci—kriptografi; enkripsi; block cipher; jaringan feistel

I. PENDAHULUAN

Kriptografi adalah ilmu yang menjaga kerahasiaan pesan dengan menyandikan pesan menjadi bentuk yang terlihat tidak bermakna. Ilmu ini sudah ada dan digunakan sejak zaman Mesir kuno dan terus berkembang hingga sekarang. Saat ini, kriptografi banyak digunakan misalnya untuk mengamankan pengiriman pesan, sistem pengamanan gedung, bahkan untuk mengamankan telepon seluler.

Pada awal kemunculannya, kriptografi hanya digunakan untuk mengenkripsi pesan sehingga tidak dapat dibaca oleh pihak lain. Namun, seiring dengan perkembangannya, kriptografi juga memberikan aspek-aspek keamanan yang lain, yaitu kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan. Perkembangan tersebut juga telah melahirkan berbagai algoritma kriptografi modern baru yang menggunakan berbagai fungsi matematika, diantaranya fungsi permutasi dani substitusi, sehingga prinsip *confusion* dan *diffusion* terpenuhi.

Makalah ini akan membahas rancangan algoritma *block cipher* baru yang memiliki tingkat kerumitan yang cukup kompleks sehingga sukar untuk dipecahkan. Algoritma bernama PIE Algorithm ini memanfaatkan algoritma vigenere cipher dan fungsi transposisi sebagai fungsi dalam jaringan feistel.

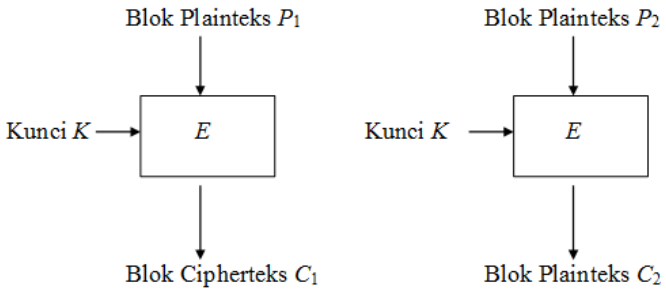
II. DASAR TEORI

A. Block Cipher

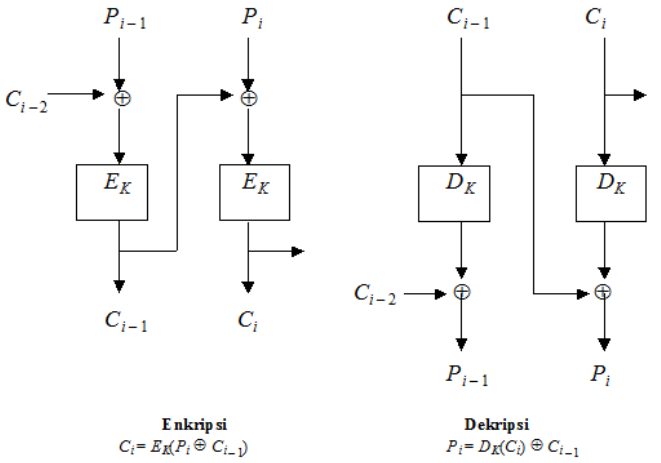
Dalam kriptografi, *block cipher* merupakan algoritma deterministik yang beroperasi pada sekelompok bit yang panjangnya telah dipastikan, disebut *block*, dengan transformasi yang tidak berubah yang dispesifikasikan dengan sebuah kunci simetri. *Block cipher* adalah komponen yang penting dalam perancangan kriptografi dan secara luas digunakan untuk implementasi enkripsi dari data yang besar.

Algoritma block cipher mencakup 2 algoritma, yaitu enkripsi (E) dan dekripsi (D). Kedua algoritma ini menerima dua input: sebuah block dengan ukuran *n-bit* dan kunci dengan ukuran k-bit, dan keduanya menghasilkan output berupa block n-bit. Algoritma dekripsi D didefinisikan sebagai invers dari fungsi enkripsi, $D = E^{-1}$.

Ada empat cara (mode) dalam mengoperasikan blok plainteks/cipherteks, yaitu *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), dan *Output Feedback* (OFB). Dalam mode ECB, setiap block plaintext P_i dienkripsi secara individual dan independen menjadi block ciphertext C_i. Dengan demikian, setiap operasi blok plainteks/cipherteks tidak akan bergantung dengan blok teks yang lain. Mode ECB ini memiliki kelemahan yaitu enkripsi suatu blok yang sama akan selalu menghasilkan blok cipherteks yang sama. Sedangkan pada mode CBC, setiap ciphertext bergantung tidak hanya pada block plainteks tetapi juga pada seluruh block plainteks sebelumnya. Hasil enkripsi block sebelumnya dilakukan umpan balik ke dalam *block* enkripsi yang sedang diproses. Namun, kebergantungan antar blok ini menyebabkan suatu operasi harus menunggu operasi blok sebelumnya selesai. Untuk mengatasinya, digunakan lah mode CFB. Pada mode CFB, operasi bit dilakukan per unit yang ukurannya lebih kecil dari ukuran blok.

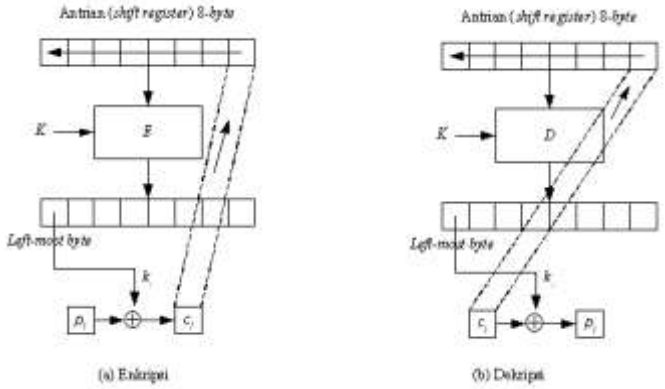


Gambar 1. Skema enkripsi dan dekripsi dengan mode ECB

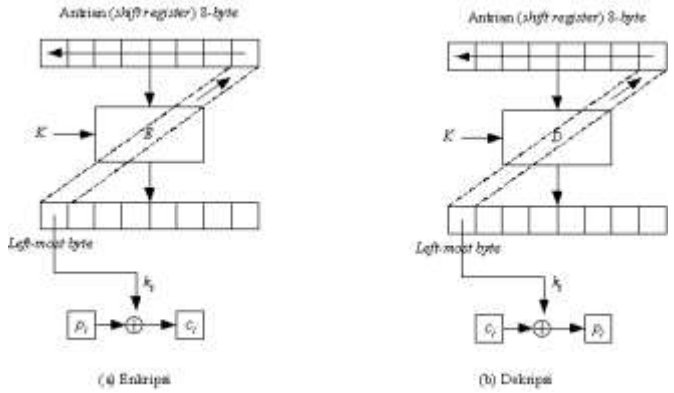


Gambar 2. Skema enkripsi dan dekripsi dengan mode CBC

Dua mode lainnya yaitu CFB dan OFB dirancang untuk memperbaiki kelemahan yang muncul pada dua mode sebelumnya. CFB mengatasi kelemahan pada mode CBC jika diterapkan pada komunikasi data (ukuran *block* yang belum lengkap). Data dienkripsikan dalam unit yang lebih kecil daripada ukuran block. Unit yang dienkripsikan dapat berupa bit per bit, 2-bit, 3-bit, dan seterusnya. Jika unit yang dienkripsi setiap kalinya hanya terdiri dari satu karakter, maka mode CFB-nya disebut CFB 8-bit. CFB n-bit melakukan enkripsi plainteks sebanyak n-bit, $n<=m$, m, menyatakan ukuran blok. Dengan kata lain, CFB melakukan operasi enkripsi *block cipher* seperti pada stream cipher. Sedangkan mode *Output Feedback* (OFB) mirip dengan mode CFB. Perbedaannya yaitu pada OFB, n-bit hasil dari enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan pada antrian.



Gambar 3. Skema enkripsi dan dekripsi dengan mode CFB



Gambar 4. Skema enkripsi dan dekripsi dengan mode OFB

B. Shannon's Confusion and Diffusion

Dalam ilmu kriptografi, istilah *confusion* dan *diffusion* merujuk kepada dua prinsip operasi penyadian (*encoding* data). Kedua prinsip ini diperkenalkan oleh Claude Shannon pada tahun 1949 dalam publikasinya yang berjudul *Communication Theory of Secrecy Systems*. Kedua prinsip tersebut bertujuan untuk menggagalkan kriptanalis dalam memecahkan *cipher* berdasarkan analisis statistik.

Dalam publikasinya, Shannon mendefinisikan *confusion* sebagai proses perancangan plainteks, kunci dan *ciphertext* yang memiliki keterhubungan serumit mungkin. Sebagai contoh, pada *cipher* substitusi seperti Caesar Cipher, hubungan antara cipherteks dan plainteks mudah diketahui, karena satu huruf yang sama pada plainteks diganti dengan satu huruf yang sama pada cipherteksnya. Prinsip confusion akan membuat seorang kriptanalis frustrasi untuk mencari pola-pola statistik yang muncul dalam cipherteks. Sedangkan *diffusion* diartikan sebagai prinsip untuk menyebarkan pengaruh bit plainteks atau kunci ke sebanyak mungkin cipherteks. Sebagai contoh, perubahan kecil pada plainteks sebanyak satu atau dua bit menghasilkan perubahan pada cipherteks yang tidak dapat diprediksi. Seperti halnya pada prinsip confusion, prinsip diffusion ini juga bertujuan untuk menyembunyikan hubungan statistik antara plainteks, kunci, dan cipherteks. Untuk memperoleh tingkat keamanan cipher yang tinggi, prinsip confusion dan diffusion diterapkan secara berulang pada sebuah blok tunggal dengan kombinasi yang berbeda-beda.

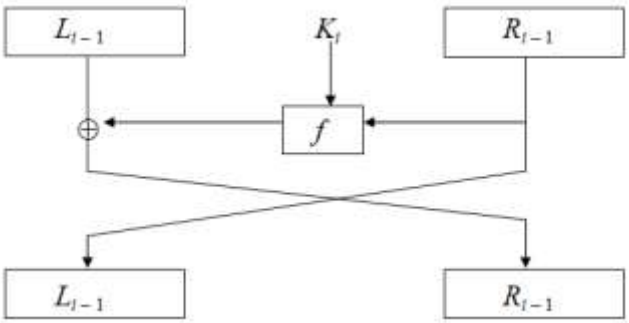
Cara yang paling sederhana agar kedua prinsip *diffusion* dan *confusion* terpenuhi adalah dengan menerapkan jaringan substitusi-permutasi. Pada penerapannya, plainteks dan kunci memiliki keterlibatan yang hampir sama dalam menghasilkan output, akibatnya, kedua prinsip diffusion dan confusion dapat terpenuhi.

C. Jaringan Feistel

Jaringan feistel adalah suatu model yang umum digunakan dalam algoritma kriptografi modern. Jaringan feistel merupakan struktur simetris yang digunakan dalam konstruksi dari block cipher. Nama *feistel* berasal dari nama fisikawan Jerman sekaligus kriptografer Horst Feistel yang menjadi pelopor riset ketika bekerja untuk IBM(USA). Struktur feistel

memiliki keuntungan yang terletak pada kemiripan operasi enkripsi dan dekripsi, bahkan identik dalam beberapa kasus, dan hanya membutuhkan kunci yang dibalik.

Cara kerja model ini yaitu pertama blok plainteks yang akan melewati jaringan feistel dibagi menjadi dua, bagian kanan dan kiri. Bagian kanan blok plainteks akan menjadi bagian kiri blok cipherteks, sementara bagian kiri blok plainteks akan di-XOR-kan dengan hasil operasi bagian kanan blok plainteks. Operasi tersebut berbeda-beda bergantung pada jenis algoritmanya. Setelah itu, bagian kiri blok plainteks yang telah di-XOR-kan akan menjadi bagian kanan dari blok cipherteks. Struktur jaringan feistel dapat dilihat pada Gambar 5 dibawah ini.

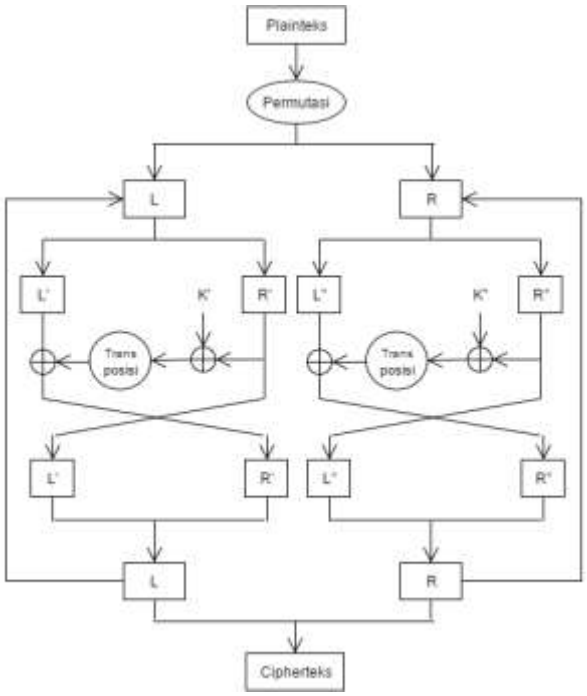


Gambar 5. Struktur jaringan feistel

Jaringan feistel ini sering digunakan karena memiliki sifat *reversible* sehingga untuk proses dekripsi, dapat digunakan algoritma yang sama dengan algoritma enkripsinya. Dalam satu kali proses enkripsi, blok bit dapat melewati jaringan feistel berkali-kali untuk menambah kerumitan proses.

III. RANCANGAN *BLOCK CIPHER*

Algoritma PIE menggunakan dua buah jaringan feistel dengan fungsi yang mirip di dalamnya. Setiap blok pesan yang akan dienkripsi akan dibagi menjadi dua bagian sama panjang dan masing-masing akan masuk ke jaringan feistel yang berbeda. Ukuran blok tersebut disamakan dengan ukuran kunci yang minimal panjangnya 64 bit. Untuk menambah kerumitan proses, blok plainteks akan dipermutasi lebih dulu sebelum dimasukkan ke dalam jaringan feistel. Secara ringkas, proses enkripsi algoritma PIE dapat dilihat pada Gambar 6.



Gambar 6. Skema enkripsi pada *PIE Algorithm*

A. Permutasi

Permutasi dilakukan pada blok plainteks sebelum blok tersebut masuk ke dalam jaringan feistel. Permutasi ini dilakukan dengan blok dalam format biner. Jadi, misalkan blok plainteks adalah 'AB' (16 bit). Blok tersebut diubah menjadi string biner yaitu 0100 0001 0100 0010.

Tabel 1. Blok plainteks

Idx	0	1	2	3	4	5	6	7
Bin	0	1	0	0	0	0	0	1

Idx	8	9	10	11	12	13	14	15
Bin	0	1	0	0	0	0	1	0

Permutasi dilakukan dengan cara memilih indeks mana (dari 0 sampai 15) yang akan diletakkan di indeks ke-i. Pemilihan indeks dilakukan secara random dengan *seed* berasal dari kunci. Misalkan bilangan random yang pertama adalah 35. 35 mod 16 = 3, maka indeks ke-0 hasil permutasi diisi dengan nilai indeks ke-3, sementara indeks ke-3 pada blok plainteks dihapus.

Tabel 2. Hasil Permutasi

Idx	0	1	2	3	4	5	…	…	15
Bin	0								

Tabel 3. Hasil Akhir Permutasi

Idx	0	1	2	3	4	5	6	7
Bin	0	1	0	0	0	0	1	0

Idx	8	9	10	11	12	13	14	15
Bin	1	0	0	0	0	1	0	0

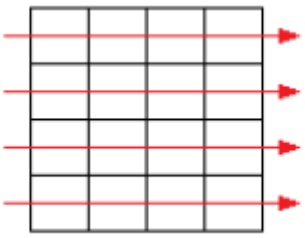
Ulangi langkah tersebut hingga seluruh tabel pada hasil permutasi terisi dan blok plainteks habis.

Selain untuk permutasi blok plainteks, cara permutasi ini juga diterapkan pada kunci. Hal ini dilakukan agar kunci yang digunakan dalam jaringan feistel selalu berbeda di setiap putarannya.

B. Jaringan Feistel

Setelah dipermutasi, blok plainteks dibagi menjadi dua bagian dengan ukuran yang sama. Masing-masing kemudian masuk ke jaringan feistel yang berbeda. Dalam setiap jaringan feistel, terdapat dua fungsi yang dioperasikan pada potongan blok plainteks.

1. Dilakukan operasi XOR terhadap potongan block plainteks dan potongan kunci. Jadi, kunci dalam bentuk biner juga dibagi menjadi dua bagian. Potongan blok plainteks yang pertama akan di-XOR-kan dengan potongan kunci yang kedua. Sedangkan potongan blok plainteks yang kedua akan di-XOR-kan dengan potongan kunci yang pertama.
2. Potongan blok plainteks ditransposisi dalam bentuk matriks. Misalkan ukuran potongan blok plainteks adalah 16 bit, maka blok tersebut akan ditempatkan dalam matriks berukuran 4 x 4. Matriks pada jaringan feistel pertama akan diputar 90° searah jarum jam, sementara matriks pada jaringan feistel kedua akan diputar 90° berlawanan arah jarum jam.



Gambar 7. Urutan peletakan dan pengambilan bit pada matriks

Masing-masing potongan blok plainteks akan melewati jaringan feistel ini sebanyak lima kali. Setelah selesai, keduanya digabungkan kembali secara terbalik, yaitu potongan kanan diletakkan di sebelah kiri dan sebaliknya.

IV. Eksperimen dan Pembahasan Hasil

Untuk pengujian algoritma PIE, kami menggunakan dua mode untuk mengoperasikan blok plainteks, yaitu ECB dan CBC. Untuk ketiga pengujian tersebut, digunakan kunci berukuran 96 bit dan plainteks berukuran 54x8 = 432 bit. Jadi, ada 5 blok plainteks yang masing-masing berukuran 96 bit. Kedua pengujian tersebut menghasilkan cipherteks yang berbeda karena adanya perbedaan cara pengoperasian blok plainteks. Namun, bagian awal cipherteks kedua pengujian tersebut sama, karena operasi blok pertama pada kedua mode tersebut sama.

Tabel 4. Hasil pengujian dengan mode ECB

Kunci	piealgorithm
Plainteks	try dan pipin makan nasi ayam pecel hingga kekenyangan
Cipherteks	GhØª»ü¥èOOëOkÄOêOz[:z1|èëiOOI Ý|OOWÿm¹ÖOUOeÞOÏè¬3*OO-n

Tabel 5. Hasil pengujian dengan mode CBC

Kunci	piealgorithm
Plainteks	try dan pipin makan nasi ayam pecel hingga kekenyangan
Cipherteks	GhØª»ü¥èŠ˜ë□¬fJ— pè‡˜¿š‡□¹_ìÉ¹Úï¢»¾□ú4#Á*´□si*P□Đ¢Â

Jika sebagian plainteks pada pengujian tersebut diubah, maka hasil yang didapatkan adalah sebagai berikut.

Tabel 6. Hasil pengujian dengan mode ECB

Kunci	piealgorithm
Plainteks	try dan pipin makan nasi sapi pecel hingga kekenyangan
Cipherteks	GhØª»ü¥èOOëOkÄOêOz[:z1|èëMO% $N}¾Đ£O¿m¹ÖOUOeÞOÏè¬3*OO-n

Tabel 7. Hasil pengujian dengan mode CBC

Kunci	piealgorithm
Plainteks	try dan pipin makan nasi sapi pecel hingga kekenyangan
Cipherteks	GhØª»ü¥èOOëO¬fJOpèOO¿OO(@O2 ¨i{OO|ûn®O2F³+OOOÅEOIOñüv

Pada mode ECB, sebagian cipherteks di bagian tengah mengalami perubahan. Hal ini disebabkan karena kata ‘ayam’ pada plainteks diganti dengan ‘sapi’. Namun, bagian akhir cipherteks tetap sama, karena enkripsi blok pada mode ini

tidak saling bergantung dengan enkripsi blok sebelumnya. Sebaliknya, pada mode CBC, perubahan cipherteks terjadi mulai dari bagian tengah hingga akhir. Hal ini disebabkan karena perubahan satu blok akan mempengaruhi blok-blok selanjutnya.

V. ANALISIS KEAMANAN

Berikut ini merupakan hasil eksperimen yang dilakukan terhadap *plaintext* yang dienkripsi menggunakan algoritma PIE yang telah dirancang.

A. Analisis Frekuensi

Dalam kriptografi, salah satu serangan yang umum digunakan untuk memecahkan cipherteks adalah dengan analisis frekuensi. Hal ini mudah dilakukan jika huruf yang sama selalu dienkripsi menjadi karakter yang sama. Dengan menggunakan algoritma PIE, setiap karakter dapat dienkripsi menjadi karakter yang berbeda-beda karena adanya permutasi kunci yang mengakibatkan kunci selalu berubah.

Tabel 8. Hasil Enkripsi Plainteks

Kunci	cryptography
Plainteks	a brown fox jumps over another brown fox
Cipherteks	?Â°øW!¼&/,ÍÂÎ□ì„Ç □Jâ□!k♯ züõfRÒÞ□□[□´ª~

Dalam percobaan tersebut, terdapat kata yang berulang pada plainteks, yaitu “brown fox”. Namun, pada cipherteks tidak ada teks yang berulang sehingga sulit untuk dilakukan analisis frekuensi.

Pengujian selanjutnya dilakukan dengan melakukan analisis frekuensi terhadap blok yang berulang pada plainteks. Pengujian dilakukan menggunakan kunci enkripsi sepanjang 64 bit. Plainteks yang digunakan terdiri dari 128 bit yang akan dipecah ke dalam 2 blok yang masing-masing memiliki bit identik dengan panjang 64.

Tabel 9. Hasil Enkripsi 2 Blok Identik

Mode	ECB	CBC
Kunci	icapipin	icapipin
Plainteks	darmawandarmawan	darmawandarmawan
Cipherteks	"(öwÕ*×«Œ´^dÀéë □	"(öwÕ*×«Íî‡£ä□‹›

Berdasarkan tabel diatas, blok 1 (darmawan) dan blok 2 (darmawan) akan dienkripsi menjadi cipherteks yang saling berbeda baik pada mode ECB maupun CBC. Hal ini dikarenakan adanya penerapan operasi permutasi terhadap plainteks sehingga posisi setiap karakter pada masing-masing yang akan masuk ke jaringan feistel akan berbeda dari posisi awal. Dengan demikian, algoritma PIE ini bebas dari bentuk serangan analisis frekuensi terhadap blok.

B. Perbedaan Kunci Enkripsi

Teknik kriptanalisis lain untuk memecahkan sebuah cipherteks adalah dengan menggunakan kunci yang berdekatan, kemudian dilihat hasil enkripsinya apakah dapat ditemukan suatu pola atau kemiripan dengan hasil enkripsi pesan yang sama sebelumnya. Percobaan dilakukan dengan mengganti 1 karakter terakhir pada kunci. Dapat dilihat pada tabel dibawah ini bahwa pesan yang dihasilkan sangat berbeda dan tidak ditemukan pola atau kemiripan yang dapat dianalisis lebih lanjut. Hal ini disebabkan oleh penerapan prinsip diffusion yang didapatkan melalui mekanisme permutasi dan transformasi kunci. Dengan demikian, algoritma PIE ini bebas dari bentuk serangan analisa kunci berdekatan.

Tabel 10. Perbedaan Hasil Enkripsi dengan Perubahan Kunci

Kunci	cryptography	cryptographi
Plainteks	a brown fox jumps over another brown fox	a brown fox jumps over another brown fox
Cipherteks	?Â°øW!¼&/,ÍÂÎ□ì„Ç □Jâ□!k# züõfRÒÞ,[□´ª~	¬l□@¦["àF0†D□õ□ □$›□dúyªuÂ4? Ó°>£µ³®~=9:w

C. Perbedaan Karakter Plainteks

Selain dengan mengubah satu karakter pada kunci, keamanan algoritma PIE juga dapat dianalisis dengan cara melakukan sedikit perubahan pada plainteks. Pengujian dilakukan dengan mengubah satu karakter pada plainteks yang kemudian dienkripsi menggunakan kunci yang sama. Hasil pengujian dapat dilihat pada tabel dibawah ini.

Tabel 11. Perbedaan Hasil Enkripsi dengan Perubahan Plainteks

Kunci	cryptography	cryptography
Plainteks	a brown fox jumps over another brown fox	a brown fix jumps over another brown fox
Cipherteks	?Â°øW!¼&/,ÍÂÎ□ì□ç □Jâ□!k♯ züõfRÒÞ□□[□´ª~	=Á°úVa@& ÉêÎ ì„Ç Jâ !k♯ züõfRÒÞ, [´ª~

Dari hasil pengujian tersebut, terlihat bahwa perbedaan kedua cipherteks cukup signifikan. Perubahan satu karakter pada plainteks hanya menyisakan beberapa kesamaan karakter pada hasil enkripsi kedua plainteks. Sehingga dapat disimpulkan bahwa algoritma ini cukup baik jika digunakan untuk melakukan enkripsi pesan yang hampir sama secara berulang.

D. Serangan *Brute Force*

Keamanan algoritma PIE salah satunya bergantung dengan panjang kunci. Makin panjang kunci, maka makin lama pula waktu yang dibutuhkan untuk menerka kunci sehingga keamanan pesan makin terjaga. Panjang kunci minimal dalam algoritma ini adalah 64 bit, yang berarti bahwa ada 2^{64} = 67108864 kemungkinan kunci. Jika dalam satu detik komputer dapat melakukan 10^6 percobaan, maka diperlukan waktu 5,86 x 10^{11} tahun untuk menemukan kunci. Waktu ini akan terus bertambah lama karena panjang kunci dapat lebih dari 64 bit.

VI. KESIMPULAN DAN SARAN

Algoritma PIE merupakan algoritma enkripsi dan dekripsi pesan yang digunakan untuk menjaga kerahasiaan pesan. Algoritma ini merupakan pengembangan dari konsep jaringan feistel serta penggunaan fungsi substitusi dan transposisi. Dari serangkaian eksperimen yang telah dilakukan, dapat disimpulkan bahwa algoritma ini dapat bekerja dengan baik dan memiliki tingkat keamanan yang cukup baik pula. Algoritma yang sederhana juga membuat proses komputasi tidak memakan waktu lama.

Dengan dikembangkannya algoritma ini, kami berharap algoritma PIE dapat berkontribusi dalam perkembangan ilmu kriptografi serta menginspirasi pihak-pihak lain untuk dapat mengembangkan algoritma yang lebih baik lagi. Untuk kedepannya, kami berharap algoritma PIE dapat dimodifikasi sehingga menghasilkan tingkat keamanan yang lebih tinggi.

REFERENSI

- http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2016-2017/kripto16-17.htm diakses pada 24 Oktober 2016 pukul 16:00 WIB
- http://cryptography.wikia.com/wiki/Confusion_and_diffusion diakses pada 24 Oktober pukul 16:30 WIB
- http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2016-2017/kripto16-17.htm diakses pada 24 Oktober 2016 pukul 16:00 WIB
- http://cryptography.wikia.com/wiki/Confusion_and_diffusion diakses pada 24 Oktober pukul 16:30 WIB
- https://www.cybrary.it/study-guides/comptia-casp/confusion-and-diffusion-and-their-role-in-cryptography/ diakses pada 24 Oktober pukul 16:35 WIB
- https://www.tutorialspoint.com/cryptography/feistel_block_cipher.htm diakses pada 24 Oktober pukul 16.45 WIB

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, 28 Oktober 2016

Penulis