APLIKASI KRIPTOGRAFI UNTUK KEAMANAN DATA DENGAN MENGGUNAKAN METODE PERTUKARAN KUNCI DIFFIE-HELLMAN DAN ALGORITMA RIJNDAEL

Ridwan[1], Ir. Taryana Suryana, M.Kom[2]

Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-116, Coblong, Lebakgede, Bandung, Kota Bandung, Jawa Barat 40132
E-mail : Ridwan.10109502@yahoo.com[1], taryanarx@gmail.com[2]

ABSTRAK

Kriptografi adalah ilmu yang berhubungan dengan transformasi data agar data tersebut menjadi sulit untuk di pahami dengan cara mengenkripsi, agar mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak dinginkan. kriptografi juga dapat diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang mudah dipahami. Sehingga kriptografi juga dapat diartikan sebagai proses untuk melindungi data.

Aplikasi kriptografi untuk keamanan data dengan menggunakan metode pertukaran kunci diffie-hellman dan algoritma rijndael (studi kasus di badan kepegawaian negara regional III) adalah untuk mengamankan file ketika akan melakukan pengiriman, pertama pengirim sama penerima melakukan pertukaran kunci atau generate kunci untuk file yang akan di enkripsi yaitu dengan menggunakan algoritma pertukaran kunci Diffie - Hellman, setelah mendapatkan kunci file akan di enkripsi dengan menggunakan algoritma rinjndael untuk enkripsi dan dekripsi file, pengirim akan melakukan pengiriman file ke pihak penerima melalui server terlebih dahulu untuk menympan data, kemudian penerima melakukan dekripsi file untuk membaca file tersebut supaya dapat dibaca kembali oleh si penerima.

Kata kunci : *Kriptografi*, Diffie -Hellman, Algoritma Rijndael, *Client Server*

PENDAHULUAN

Badan Kepegawaian Negara (BKN) Regional III yang berlokasi di Jln. Surapati No.10 Bandung 40124 adalah lembaga pemerintahan yang dibentuk untuk melaksanakan tugas pemerintahan di bidang manajemen kepegawaian negara sesuai dengan ketentuan peraturan perundang-undangan yang berlaku.

BKN mempunyai beberapa bagian diantaranya bagian keuangan, bagian kepegawaian, bidang mutasi, bidang informasi, dan bidang teknis kepegawaian. maka dengan banyaknya dokumen yang harus diolah di BKN regional III perlu pengamanan dokumen yang harus dikelola dengan baik supaya tidak ada penyalahgunaan data oleh pihak ketiga. Di sisi lain Pertukaran dokumen seperti berbasis komputer sudah berkembang menjadi salah satu media yang paling banyak digunakan, maka dengan media pengiriman dokumen melalui jaringan komputer.

Pada saat ini di BKN regional III menghadapi masalah dalam keamanan dokumen, dokumen bebas diakses oleh berbagai pihak dan tidak ada pengamanan dalam dokumen dokumen antara lain dokumen keuangan, kepegawaian, mutasi, dan pengangkatan sehingga orang bisa membaca dan merubah karna belum ada aplikasi untuk mengenkripsi file,maka dari itu diperlukan suatu keamanan yang baik sehingga dokumen yang terdapat pada komputer menjadi lebih aman. Banyaknya data yang harus diolah perlu pengamanan data yang harus dikelola dengan baik supaya tidak ada penyalahgunaan seperti NIP pegawai yang dipalsukan, dokumen kepegawaian yang disalah gunakan, maka di BKN harus dibuat sistem untuk mengamankan dokumen ketika mengirimkan dokumen tersebut ke penerima supaya tidak bisa dibuka dokumennya**.** Maka untuk mengamankan dokumen yang bersifat penting dan mengatasi hal-hal diatas diperlukannya sebuah cara untuk mengamankan suatu data atau informasi dengan menggunakan teknik kriptografi.

Landasan Teori

2.1 keamanan data

keamanan menrupakan salah satu aspek paling penting dalam sebuah system informasi,maslh keamanan sering sekali kurang di perhatikan, Informasi menentukan hampir setiap elemen dari setiap kehidupan manusia. Informasi sangat penting artinya bagi kehidupan karna tanpa informasi maka hampir semua tidak dapat dilakukan dengan baik. Contohnya jika kita membeli tike kereta dan membayarnya dengan menggunakan kartu kredit, informasi mengenai diri kita akan disimpan dan di kumpulkan serta digunakan oleh Bank dan

penerbangan. Hanya sedikit hal yang bias dilakukan di dunia modern tanpa melibatkan pengumpulan, penukaran,pembuatan atau pengaksesan informasi. Saat ini informasisudah menjadi komoditas yang sangat penting. Bahkan ada yang mengatakan bahwa kita sudah berada disebuah information/based society.

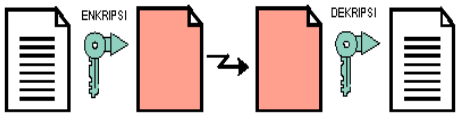
Kemampuan untuk mengakses dan menyediakan informasi secara cepat dan akurat menjadi sangat esiensial bagi sebuah organisasi, baik organisasi komersial (perusahaan), perguruan tinggi, lembaga pemerintahan, maupun organisasi social.

2.1.1 Kriptografi

Secara etimologi kata kriptografi (Cryptography) berasal dari bahasa Yunani, yaitu kryptos yang artinya yang tersembunyi dan graphein yang artinya tulisan (Prayudi, 2005). Awal mula kriptografi dipahami sebagai ilmu tentang menyembunyikan pesan (Sadikin, 2012), tetapi seiring perkembangan zaman hingga saat ini pengertian kriptografi berkembang menjadi ilmu tentang teknik matematis yang digunakan untuk menyelesaikan persoalan keamanan berupa privasi dan otentikasi (Diffie, 1976). Sejarah penulisan rahasia tertua dapat ditemukan pada peradaban Mesir kuno, yakni tahun 3000 SM. Bangsa Mesir menggunakan ukiran rahasia yang disebut dengan hieroglyphics untuk menyampaikan pesan kepada orang-orang yang berhak.

kata kriptografi berasal dari gabungan dua kata dalam bahasa Yunani yaitu “kriptos” dan “graphia”. Kata kriptos digunakan untuk mendeskripsikan sesuatu yang disembunyikan, rahasia atau misterius. Sedangkan kata graphia berarti tulisan. Kriptografi didefinisikan sebagai ilmu dan pelajaran untuk tulisan rahasia dengan pertimbangan bahwa komunikasi dan data dapat dikodekan untuk mencegah dari mata-mata atau orang lain yang ingin mengetahui isinya, dengan menggunakan kode-kode dan aturan-aturan tertentu dan metode lainnya sehingga hanya orang yang berhak yang dapat mengetahui isi pesan sebenarnya. Dalam arti lain, cryptography adalah seni dan ilmu dalam mengamankan pesan.

Dalam dunia kriptografi, pesan disebut plaintext atau cleartext. Proses untuk menyamarkan pesan dengan cara sedemikian rupa untuk menyembunyikan isi aslinya disebut enkripsi. Pesan yang telah dienkripsi disebut ciphertext. Proses pengembalian sebuah ciphertext ke plaintext disebut dekripsi.



Gambar 1.1 Konsep Dasar dari Enkripsi dan Dekripsi

2.2.1 DIFFIE HELMAN

Diffie-Hellman key exchange adalah metode dimana subyek menukar kunci rahasia melalui media yang tidak aman tanpa mengekspos kunci. Metode ini diperlihatkan oleh Dr. W. Diffie dan Dr. M. E. Hellman pada tahun 1976 pada papernya “New Directions in Cryptography”. Metode ini memungkinkan dua pengguna untuk bertukar kunci rahasia melalui media yang tidak aman tanpa kunci tambahan. Metode ini memiliki dua parameter sistem, p dan g.

Kedua parameter tersebut publik dan dapat digunakan oleh semua pengguna sistem. Parameter p adalah bilangan prima, dan paramater g (sering disebut generator) adalah integer yang lebih kecil dari p yang memiliki properti berikut ini: Untuk setiap bilangan n antara 1 dan p-1 inklusif, ada pemangkatan k pada g sehingga gk = n mod p Penggunaan Algoritma Diffie-Hellman dalam pertukaran kunci dapat dilakukan secara aman dan efektif dalam pemrosesan jika dibandingkan dengan algoritma RSA yang cenderung lebih lama dalam pemrosesan algoritmanya. Proses pertukaran kunci ini dapat dilakukan lebih dari 2 orang asal memenuhi 2 prinsip yang telah dibahas tadi. Algoritma Diffie-Hellman lebih memfokuskan dalam perubahan nilai kunci dan proses matematis dalam penentuan kunci akhir yang sama.

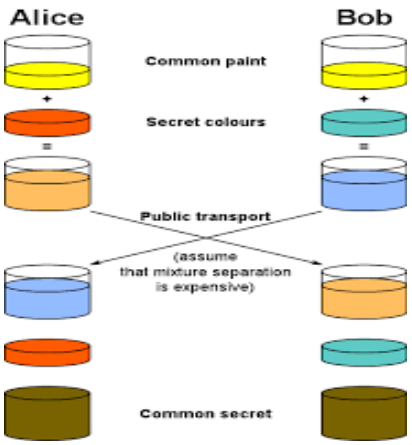
2.2.1.1 Cara Kerja Protokol Pertukaran Kunci Diffie Hellman

Protocol kesepakatan kunci Diffie-Hellman (Diffie-Hellman Key Agreement) membangun kunci sesi tanpa menggunakan kunci yang dibagikan sebelumnya. Pesan-pesan yang ditukarkan antara Alice dan Bob bisa dibaca oleh siapapun yang bisa menyadap/menguping(eavesdrop) dan si penyadap tidak akan bisa mengetahui kunci sesi yang akhirnya digunakan oleh Alice dan Bob. Di sisi lain, Diffie-Hellman tidak melakukan otentikasi pada pihak-pihak yang terlibat. Salah satu dari kegunaan utama dari DIffie-Hellman adalah dalam protokol Internet Key Exchange(IKE/ Pertukaran Kunci Internet), sebuah bagian pusat dari arsitektur IP Security(IPsec) Protokol Diffie-Hellman mempunyai dua parameter, p dan g, yang mana keduanya bersifat publik dan mungkin akan digunakan oleh semua user dalam sistem tertentu.

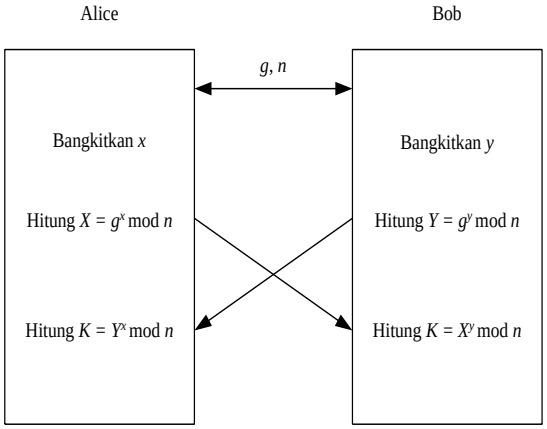
Parameter p harus berupa bilangan prima. Bilangan-bilangan bulat mod p (singkatan dari modulo p) mempunyai jangkauan dari 0 hingga p-1,

karena x mod p adalah sisa dari x dibagi p, dan membentuk apa yang para matematikawan sebut sebagai group dibawah perkalian. Parameter g (biasanya disebut sebuah generator) harus berupa sebuah akar primitive(primitive root) dari p: untuk setiap angka n dari 1 sampai p -1 pasti ada nilai k sehingga n = g^k mod p. sebagai contoh jika p adalah bilangan prima 5 (pada sistem yang sebenarnya akan digunakan angka yang jauh lebih besar), maka kita akan memilih 2 sebagai generator karena:
1 = 2^0 mod p
2 = 2^1 mod p
3 = 2^3 mod p
4 = 2^2 mod p

Misalkan Alice dan Bob ingin sepakat pada sebuah kunci simetris yang terbagi. Alice dan Bob dan orang lain sudah mengetahui nilai p dan g. Alice meng-generate sebuah nilai privat acak a dan Bob meng-generate sebuah nilai acak privat b. baik a dan b diambil dari sekumpulan integer {1,….,p-1}. Alice dan Bob menurunkan pasangan nilai publik mereka, nilai yang akan mereka kirimkan satu sama lain tanpa terenkripsi, sebagai berikut. Nilai publik dari Alice adalah :
g^a mod p dan nilai publik Bob adalahg^b mod p
Kemudian mereka menukar nilai pubik mereka. Akhirnya, Alice menghitung
g^ab mod p = (g^b mod p)^a mod p
dan bob menghitung
g^ba mod p = (g^a mod p)^b mod p
Alice dan Bob sekarang mempunyai gab mod p = g^ba mod p sebagai kunci simetris mereka yang dibagikan satu sama lain.Sebuah varian dari Diffie-Hellman terkadang disebut Diffie-Hellman yang sudah diperbaiki(fixed) memungkinkan otentikasi dari satu atau kedua pihak.



Gambar 1.2 pertukaran kunci Diffie- Hellman



Gambar 1.3 Proses Diffie-Helman

Contoh 1 :

Alice dan Bob menyepakati $n = 97$ dan $g = 5$ ($g < n$)

1. Alice memilih $x = 36$ dan menghitung
   $X = g^x \bmod n = 5^{36} \bmod 97 = 50$
   Alice mengirimkan X kepada Bob.
2. Bob memilih y = 58 dan menghitung
   $Y = g^y \bmod n = 5^{58} \bmod 97 = 44$
   Bob mengirimkan Y kepada Alice.
3. Alice menghitung kunci simetri K,
   $K = Y^x \bmod n = 44^{36} \bmod 97 = 75$
4. Bob menghitung kunci simetri K,
   $K = X^y \bmod n = 50^{58} \bmod 97 = 75$

Jadi, Alice dan Bob sekarang sudah mempunyai kunci enkripsi simetri yang sama, yaitu $K = 75$.

2.2.2 ALGORITMA RIJNDAEL

Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan cipher block. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Rijndael mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun Rijndael mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

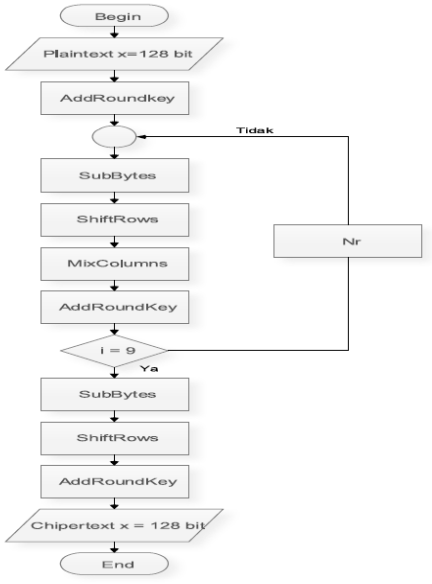
Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan cipher block. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Rijndael mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun Rijndael mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit.

Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk

proses enkripsi dan dekripsi. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

Blok-blok data masukan dan kunci dioperasikan dalam bentuk array. Setiap anggota array sebelum menghasilkan keluaran ciphertext dinamakan dengan state. Setiap state akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, AddRoundKey, SubBytes, ShiftRows, dan MixColumns. Kecuali tahap MixColumns, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap MixColumns tidak akan dilakukan pada tahap terakhir. Proses dekripsi adalah kebalikkan dari dekripsi. Karena terjadi beberapa tahap dalam proses enkripsi, maka diperlukan subkeysubkey yang akan dipakai pada tiap tahap.

Pengembangan jumlah kunci yang akan dipakai diperlukan karena kebutuhan subkeysubkey yang akan dipakai dapat mencapai ribuan bit, sedangkan kunci yang disediakan secara default hanya 128-256 bit. Jumla total kunci yang diperlukan sebagai subkey adalah sebanyak Nb(Nr+1), dimana Nb adalah besarnya blok data dalam satuan word. Sedangkan Nr adalah jumlah tahapan yang harus dilalui dalam satuan word. Sebagai contoh, bilamana digunakan 128 bit (4 word) blok data dan 128 bit (4 word) kunci maka akan dilakukan 10 kali proses (lihat Tabel 1). Dengan demikian dari rumus didapatkan 4(10+1)=44 word=1408 bit kunci. Untuk melakukan pengembangan jumlah kunci yang akan dipakai dari kunci utama maka dilakukan key schedule.



Gambar 1.4 Proses Algoritm Rijndael

Pada proses enkripsi file terdapat tugas-tugas seperti berikut :

1. *Load file* teks yang akan dienkripsi. Format teks yang digunakan adalah *.txt, .doc,* dan *.pdf*. Sedangkan untuk ukuran *file*nya tidak ada batasannya, semakin besar ukuran *file*nya maka semakin banyak pula pesan yang dapat dienkripsi.
2. Pada proses enkripsi dan dekripsi *Rijndael* proses *Key Schedule* sama, mengambil kunci *cipher* dan melakukan rutin ekspansi kunci (*key expansion*) untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb(Nr+1) *word*. Algoritma ini membutuhkan set awal *key* yang terdiri dari Nb *word*, dan setiap *round* Nr membutuhkan data kunci sebanyak Nb *word*. Hasil *key schedule* terdiri dari *array* 4 *byte word* linear yang dinotasikan dengan [wi]. *SubWord* adalah fungsi yang mengambil 4 *byte word* input dan mengaplikasikan S-Box ke tiap-tiap data 4 *byte* untuk menghasilkan *word output*. Fungsi *RotWord* mengambil *word* [a0, a1, a2, a3] sebagai *input*, melakukan permutasi siklik, dan mengembalikan *word* [a1,a2,a3,a0].

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Gambar 1.5 *Rcon*

Dari Gambar 1.5 dapat dilihat bahwa *Rcon*[i] terdiri dari nilai-nilai yang diberikan oleh [xi-1, {00}, {00}, {00}], dengan xi-1 sebagai pangkat dari x (x dinotasikan sebagai {02}).

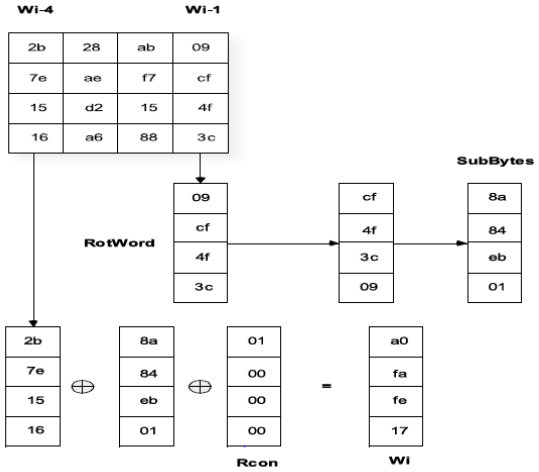
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Gambar1.6 *Cipher Key*

Dari Gambar 1.6 dapat dilihat bahwa kunci kriptografi yang digunakan oleh rutin untuk Ekspansi Kunci menghasilkan satu set *round Keys*, dapat digambarkan sebagai *array* persegi panjang *byte*, yang memiliki empat baris dan kolom Nk.

Dari Gambar 6 dapat dilihat bahwa kunci kriptografi yang digunakan oleh rutin untuk Ekspansi Kunci menghasilkan satu set *round Keys*, dapat digambarkan sebagai

array persegi panjang *byte*, yang memiliki empat baris dan kolom Nk.



Gambar 1.7 Proses *Key Schedule*

Dari Gambar 1.7 dapat dilihat bahwa *word* ke *Nk* pertama pada ekspansi kunci berisi kunci *cipher*. Setiap *word* berikutnya, w[i], sama dengan XOR dari *word* sebelumnya, w[i-1] dan *word Nk* yang ada pada posisi sebelumnya, w[i-Nk]. Untuk *word* pada posisi yang merupakan kelipatan *Nk*, sebuah transformasi diaplikasikan pada w[i-1] sebelum *XOR*, lalu dilanjutkan oleh *XOR* dengan konstanta*round*, *Rcon*[i]. Transformasi ini terdiri dari pergeseran siklik dari *byte* data dalam suatu *word RotWord*, lalu diikuti aplikasi dari *lookup* tabel untuk semua 4 *byte* data dari *word SubWord*.

3. Selama kalkulasi *plainteks* menjadi *cipherteks*, status sekarang dari data disimpan di dalam *array of byte* dua dimensi, *state*, yang berukuran *NROWS x NCOLS*. Elemen *array state* diacu sebagai berikut :

   s[r,c] = in[r+4c] untuk $0 \le r < 4$ dan $0 \le c <$ Nb

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

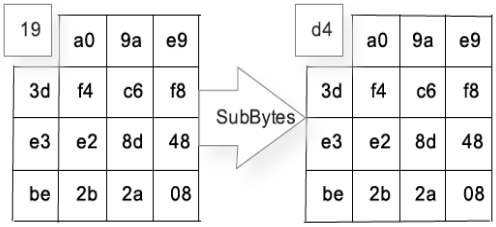
Gambar 1.8 Pengisian *Array State*

Dari Gambar 1.8 dapat dilihat bahwa(Nb adalah panjang blok dibagi 32). Pada AES, Nb = 128/32 = 4. Tiap elemen dari *array state* diisi dengan 8 *bit* teks (1 *byte*) dalam notasi *HEX*.

4. Setelah pengisian *array state* selesai, lakukan transformasi *SubBytes* dimana transformasi *byte* pada setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (*S-Box*). Tabel substitusi *S-Box* akan dipaparkan dalam Tabel 3.2.

hex	y 0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x 0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

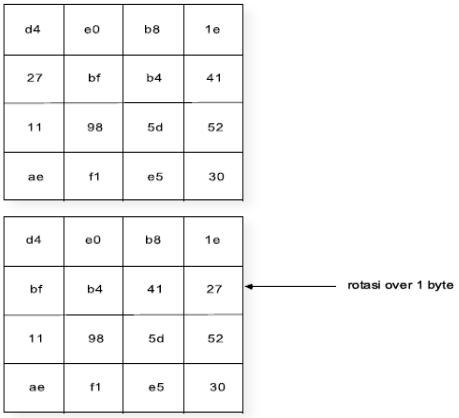
Berikut hasil *array state* yang sudah di transformasi *SubBytes* :

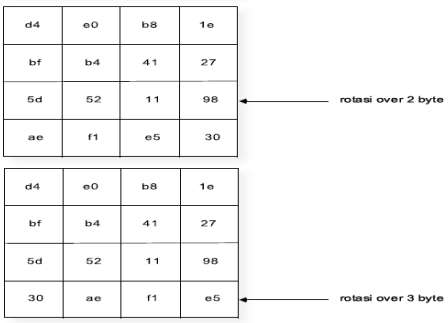


Gambar 1.9 Transformasi *SubBytes*

Dari Gambar 1.9 dapat dilihat bahwa *S-box* yang digunakan dalam *SubBytes*() transformasi disajikan dalam bentuk heksadesimal pada Tabel 3.1. Sebagai contoh, jika = 1,9s {D4}, maka nilai substitusi akan ditentukan oleh perempatan dari baris dengan indeks '1' dan kolom dengan indeks '9'.

5. Lakukan transformasi *Shiftrows* yang pada dasarnya adalah proses pergeseran *bit* dimana *bit* paling kiri akan dipindahkan menjadi *bit* paling kanan (rotasi *bit*). Proses pergeseran *ShiftRows* ditunjukan dalam Gambar 3.24 berikut :

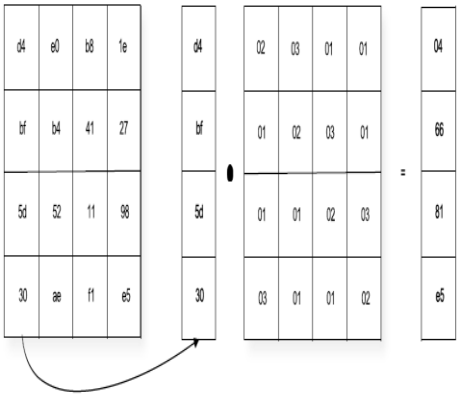




Gambar 1.10 Transformasi *ShiftRows*

6. Setelah proses transformasi *ShiftRows* selesai, lakukan pengacakan *array state* dengan cara melakukan perkalian matriks yang merupakan transformasi dari perkalian polinom antara tiap kolom dengan polinom 4 suku pada *GF(28),*

$a(x) \bmod (x4 + 1)$.



Gambar 1.11 *MixColumns*

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Gambar 1.12 *Galois Field By a Given Matrix*

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

Gambar 1.13 Hasil Akhir *MixColumns*

7. Pada proses enkripsi dan dekripsi *Rijndael* proses *AddRoundKey* sama, sebuah *round key* ditambahkan pada *state* dengan operasi *XOR*. Setiap *round key* terdiri dari *Nb word* dimana tiap word tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga :
[S'0,c,S'1,c,S'2,c,S'3,c] =
[S0,c,S1,c,S2,c,S3,c] [Wround*Nb+c]
untuk $0 \leq c \leq Nb$
[Wi] adalah word dari key yang bersesuaian dimana :

$$i = round*Nb+c$$

Transformasi *AddRoundKey* pada proses enkripsi pertama kali pada *round* = 0 untuk *round* selanjutnya *round*= *round* + 1.

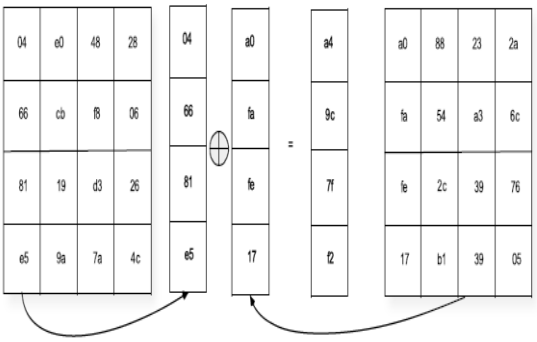
04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

Gambar 1.14 *Array State* Sekarang

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

Gambar 1.55 *Round Key*

Berikut hasil dari *array state* dan *round key* yang sudah di transformasi *AddRoundKey*:



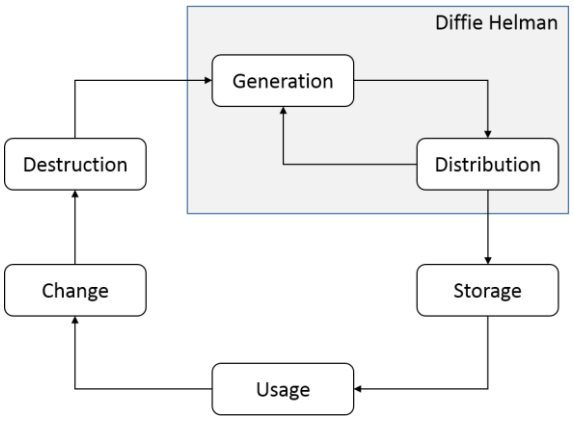


Gambar 1.16 Transformasi *AddRoundKey*

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

Gambar 1.17 Hasil Akhir *AddRoundKey*

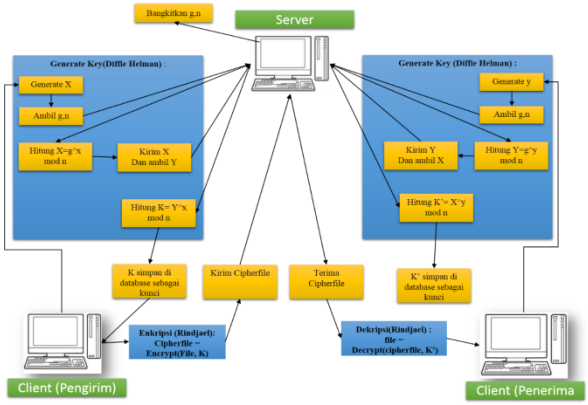
8. Setelah selesai melakukan semua langkah diatas maka simpan *file* tersebut sebagai *cipherteks* (*file* teks yang sudah terenkripsi).



Gambar 3.18 Manajemen Kunci Sistem

Terdapat 6 proses utama dimana 2 proses adalah penerapan dari algoritma Diffie Helman yaitu proses *Generation* dan *Distribution.* Sisanya yaitu proses *Storage, usage, change* dan *Destruction.*

A. *Generation* (pembangkitan)
   Proses ini merupakan impelementasi dari algoritma Diffie Helman. Variable yang dibangkitkan yaitu : variable *g* dan *n*, $X = g^x$ mod *n*, $Y = g^y$ *mod n*, $K = Y^x$ *mod n* serta $K' = X^y$ mod *n.*
B. *Distribution* (penyebaran)
   Proses ini merupakan bagian dari implementasi algoritma Diffie Helman. Variable yang di sebarkan yaitu variable *X* dan *Y.*
C. *Storage* (Penyimpanan)
   Proses ini merupakan tahap dimana setelah kunci privat didapatkan. Kunci privat lalu disimpan di database server dan diamankan.
D. *Usage* (pemakaian)
   Proses ini merupakan tahap penggunaan kunci yang diambil dari database untuk proses enkripsi dan dekripsi.
E. *Change* (Penggantian)
   Proses ini merupakan proses untuk melakukan penggantian kunci jika client merasa perlu untuk mengganti kunci dengan melakukan proses *Generation* kembali.
F. *Destruction* (Penghancuran)
   Proses ini berjalan ketika client sudah mengganti. Kunci yang berada di database dihapus.



Gambar 1.19 proses kerja system

Tahapan-tahapan yang terdapat pada Gambaran umum sistem terbagi menjadi 3 proses utama, yaitu . Tahapan- tahapan setiap prosesnya yaitu sebagai berikut :

1. Proses Generate Key
   a. Server membangkitkan bilangan *g* dan *n*.
   b. Client (Pengirim) membangkitkan bilangan *x* sekaligus mengambil bilangan *g* dan *n* dari server. Selanjutnya menghitung $X= g^x \bmod n$. Nilai *X* yang dihasilkan dikirim ke server.
   c. Client (Penerima) membangkitkan bilangan *y* sekaligus mengambil bilangan *g* dan *n* dari server. Selanjutnya menghitung $Y= g^Y \bmod n$. Nilai *Y* yang dihasilkan dikirim ke server.
   d. Client (Pengirim) mengambil nilai *Y* yang dikirim oleh client penerima, dan menghitung $K=Y^x \bmod n$ dan menyimpan K di database sebagai kunci privat untuk mengenkripsi.
   e. Client (Penerima) mengambil nilai *X* yang dikirim oleh client pengirim, dan menghitung $K'=X^Y \bmod n$ dan menyimpan K' di database sebagai kunci privat untuk mendekripsi.
2. Proses Enkripsi
   a. Client (Pengirim) menentukan alamat client (Penerima) (IP/Hostname).
   b. Client (Pengirim) memilih file di komputer dan mengambil kunci privat.
   c. Client (Pengirim) melakukan enkripsi pada file yang dipilih dengan kunci privat.
   d. Client (Pengirim) mengirimkan file ke server.
3. Proses Dekripsi
   a. Client (Penerima) memilih file di server.
   b. Client (Penerima) mengambil kunci privat.
   c. Client (Penerima) melakukan dekripi pada file yang dipilih dengan kunci privat.

Berdasarkan hasil pengujian dan evaluasi yang telah dilakukan sebelumnya, maka dapat diperoleh kesimpulan sebagai berikut: Setelah melalui tahap perancangan, implementasi dan pengujian

Aplikasi kriptografi untuk keamanan data dengan menggunakan metode pertukaran kunci diffie-hellman dan algoritma rijndael (studi kasus di badan kepegawaian negara regional III), maka dapat ditarik kesimpulan bahwa:

1. Aplikasi ini memenuhi kebutuhan yang diperlukan di BKN(Badan Kepegawaian Negara Regional III yang bertujuan untuk mengamankan Dokumen-dokumen yang dianggap penting.
2. Pada aplikasi ini file tidak akan bisa dibuka karna sudah terenkripsi dan file tersebut tidak akan bisa di salahgunakan oleh pihak yang tidak bertanggung jawab.

DAFTAR PUSTAKA

[1] Ramdan, Mangunpraja. 2010. Peningkatan Keamanan Pertukaran Kunci DiffieHellman dengan Pengimbuhan Algoritma RSA, Bandung:STIE.

[2] Kurniawan J., Ir. , M.T., *Kriptografi, Keamanan Internet dan Jaringan Komunikasi*, Informatika ,April 2004.

[3] Sadikin, Rifki. 2012. Kriptografi Untuk Keamanan Jaringan.Yogyakarta: Penerbit Andi Offset.

[4] N. Penchalaiah , "Effective Comparison and evaluation of DES and Rijndael Algorithm (AES)", International Journal on Computer Science and Engineering, Vol. 02, No. 05, 2010, pp.1641-1645.