Makalah IF3058 Kriptografi

PENGGUNAAN ENKRIPSI AES DENGAN MODE OPERASI CBC DAN CTR PADA JAVASCRIPT DENGAN LIBRARY PIDCRYPT

Alfa Pramudita
135.05.026

Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
email : if15026@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang penggunaan enkripsi dan dekripsi dalam aplikasi web yang menggunakan JavaScript dengan library PidCrypt. PidCrypt merupakan sebuah library open source dalam bahasa JavaScript untuk melakukan enkripsi maupun dekripsi terhadap suatu teks dengan sebuah kunci tertentu. Dalam library ini, terdapat beberapa algoritma enkripsi yang diimplementasikan. Algoritma yang dibahas dalam makalah ini adalah algoritma enkripsi AES yang dioperasikan dalam mode CBC serta mode CTR. AES adalah algoritma enkripsi yang merupakan pengembangan lebih lanjut dari DES. CBC dan CTR merupakan mode yang dapat digunakan untuk melakukan enkripsi cipher blok. Dengan kombinasi algoritma dan mode tersebut, hasil enkripsi teks akan lebih sulit untuk dipecahkan. Selain pembahasan implementasi algoritma tersebut, akan dibahas juga mengenai hasil enkripsi menggunakan PidCrypt dengan algoritma ini untuk beberapa kasus uji.

Kata Kunci : Advanced Encryption Standard, Rijndael, Cipher Block Chaining, Counter, Integer Counter Mode, Segmented Integer Counter, Pidder, JavaScript, PidCrypt, enkripsi, dekripsi

A. Pendahuluan

Jaringan internet yang semakin berkembang belakangan ini memberikan berbagai macam kemudahan sekaligus kekhawatiran bagi penggunanya. Semakin banyaknya layanan-layanan yang disediakan dalam jaringan tersebut sangat membantu para pengguna internet baik untuk menyelesaikan berbagai persoalan yang mereka hadapi maupun untuk sekedar menghabiskan waktu luang dalam dunia maya. Namun dibalik kemudahan-kemudahan yang diberikan oleh layanan-layanan tersebut, terdapat banyak bahaya yang mungkin dihadapi oleh para pengguna internet maupun penyedia layanan itu sendiri.

Salah satu bahaya yang kerap diperbincangkan adalah bahaya keamanan terhadap data-data yang tersimpan dalam jaringan tersebut. Data-data yang terkadang bersifat rahasia tersebut seringkali disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab. Oleh karena itu, diperlukan penanganan khusus terhadap keamanan penyimpanan data-data tersebut.

Salah satu cara untuk melindungi data-data yang tersimpan dalam jaringan internet adalah dengan cara melakukan enkripsi terhadap data-data tersebut sebelum disimpan dalam jaringan. Enkripsi merupakan sebuah prosedur untuk menyamarkan pesan maupun data sehingga tidak sembarang orang dapat menggunakan data tersebut.

Terdapat berbagai cara untuk mengimplementasikan fungsi enkripsi pada aplikasi web. Untuk para pengembang aplikasi web yang menggunakan JavaScript, implementasi fitur enkripsi dapat dipermudah dengan menggunakan library PidCrypt. PidCrypt merupakan library untuk JavaScript yang berisikan sekumpulan fungsi-fungsi enkripsi yang diimplementasikan dalam bahasa JavaScript. Penggunaan enkripsi pada sisi client akan lebih memperkuat pengamanan data karena data yang dikirimkan melalui jaringan internet telah dienkripsi terlebih dahulu.

Salah satu algoritma yang terdapat dalam library PidCrypt ini adalah AES (Advanced Encryption Standard) dengan mode CBC (Cipher Block Chaining) dan dengan mode CTR (Counter). AES merupakan pengembangan dari DES (Data Encryption Standard) dan merupakan salah satu algoritma enkripsi yang sering digunakan dalam

implementasi algoritma kunci simetri.

B. Penjelasan PidCrypt

Seperti dijelaskan sebelumnya, PidCrypt adalah sebuah library untuk JavaScript yang dapat digunakan untuk mengimplementasikan fitur enkripsi pada aplikasi web yang menggunakan JavaScript. Dalam bab ini akan dijelaskan beberapa fungsi yang diimplementasikan dalam PidCrypt serta informasi umum tentang library PidCrypt.

PidCrypt merupakan library JavaScript open source yang dikembangkan oleh Pidder. Halaman web yang berisi informasi-informasi tentang PidCrypt dapat dibuka pada alamat http://www.pidder.com/pidcrypt. Penggunaan serta lisensi PidCrypt mengikuti aturan yang dibuat oleh ICS license. Source code PidCrypt dapat diunduh pada situs web http://www.sourceforge.net secara gratis.

Fungsi-fungsi yang diimplementasikan pada PidCrypt berdasarkan jenisnya antara lain :

Encoding :
- Base64
- UTF-8

Hashing :
- MD5
- SHA-1
- SHA-256

Parsing :
- ASN.1

Symmetric Encryption :
- AES-CBC
- AES-CTR

Assymetric Encription :
- RSA

Sedangkan beberapa fungsi yang sedang dalam pengembangan dan akan datang dalam waktu dekat antara lain adalah DES-3, Serpent, dan TwoFish.

Pada makalah ini, algoritma yang akan dibahas adalah algoritma AES (Advanced Encryption Standard) dengan mode CBC (Cipher Block Chaining) serta dengan mode CTR (Counter).

C. Advanced Encryption Standard

Advanced Encryption Standard (AES) merupakan pengembangan dari DES (Data Encryption Standard). Ketika DES sudah dianggap tidak memenuhi standar keamanan, NIST (National Institute of Standard Technology) mengadakan sayembara untuk mencari pengganti DES. Dari 15 proposal yang masuk, akhirnya algoritma Rijndael dipilih sebagai algoritma AES.

Seperti halnya DES, AES juga menggunakan fungsi substitusi dan permutasi serta sejumlah putaran yang masing-masing menggunakan kunci internal yang berbeda. Salah satu yang membedakan AES dengan DES adalah operasi AES yang berorientasi byte, tidak seperti DES yang berorientasi bit. Selain itu, jaringan Feistel yang digunakan dalam DES tidak dipergunakan lagi dalam AES. Sebagai gantinya, AES menggunakan struktur SPN yang memiliki derajat paralelisme lebih besar.

Garis besar algoritma Rijndael yang beroperasi blok 128-bit dengan kunci 128-bit adalah sebagai berikut:

1. AddRoundKey: melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga initial round.

2. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. ByteSub: substitusi byte dengan menggunakan tabel substitusi (S-box). Tabel substitusi dapat dilihat pada tabel 1, sedangkan ilustrasi ByteSub dapat dilihat pada gambar 2.
   b. ShiftRow: pergeseran baris-baris array state secara wrapping. Ilustarsi ShiftRow dapat dilihat pada gambar 3.
   c. MixColumn: mengacak data di masing-masing kolom array state. Ilustarsi MixColumn dapat dilihat pada gambar 4.
   d. AddRoundKey: melakukan XOR antara state sekarang dengan round key. Ilustarsi AddRoundKey dapat dilihat pada gambar 5.

3. Final round: proses untuk putaran terakhir:
   a. ByteSub.
   b. ShiftRow.
   c. AddRoundKey.

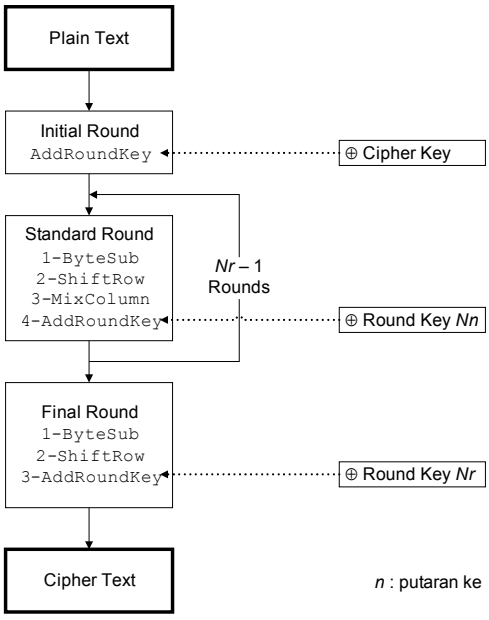
Diagram proses enkripsi AES dapat dilihat pada Gambar 1.

Algoritma Rijndael mempunyai 3 parameter sebagai berikut:
- Plainteks : array yang berukuran 16 byte, yang berisi data masukan.

– Cipherteks : array yang berukuran 16 byte, yang berisi hasil enkripsi.
– Key : array yang berukuran 16 byte, yang berisi kunci ciphering (disebut juga cipher key).

Dengan 16 byte, maka baik blok data dan kunci yang berukuran 128-bit dapat disimpan di dalam ketiga array tersebut (128 = 16 x 8).

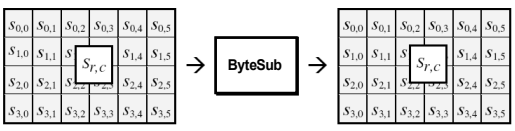
Selama kalkulasi plainteks menjadi cipherteks, status sekarang dari data disimpan di dalam array of byte dua dimensi, state, yang berukuran NROWS x NCOLS. Elemen array state diacu sebagai S[r,c], dengan $0 \le r < 4$ dan $0 \le c < Nc$ (Nc adalah panjang blok dibagi 32). Pada AES, Nc = 128/32 = 4.



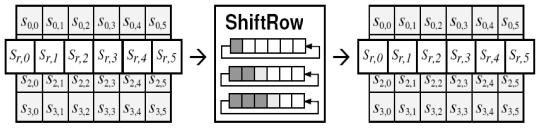
Gambar 1 Diagram Proses Enkripsi AES

Tabel 1 Tabel S-box yang digunakan dalam transformasi ByteSub() AES

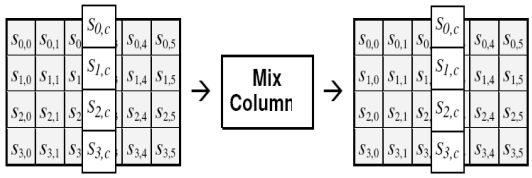
hex	y=0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x=0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	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	03	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



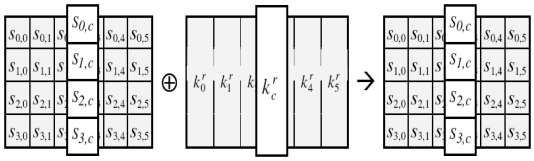
Gambar 2 Ilustrasi Transformasi ByteSub() AES



Gambar 3 Ilustrasi Transformasi ShiftRow() AES



Gambar 4 Ilustrasi Transformasi MixColumn() AES



Gambar 5 Ilustrasi Transformasi AddRoundKey() AES

D. Mode Operasi Cipher Blok

Dalam PidCrypt, terdapat dua mode operasi cipher blok yang dapat digunakan dalam algoritma AES, yaitu CBC dan CTR. Sebelum menjelaskan kedua mode tersebut, akan dijelaskan terlebih dahulu mengenai cipher blok.

Pada cipher blok, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang sama. Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci (yang ukurannya sama dengan blok plainteks). Algoritma enkripsi menghasilkan blok cipherteks yang berukuran sama dengan blok plainteks. Dekripsi dilakukan dengan cara yang serupa seperti enkripsi.

Misalkan blok plainteks (P) yang berukuran m bit dinyatakan sebagai vektor

$$P = (p_1, p_2, ..., p_m)$$

yang dalam hal ini p_i adalah bit 0 atau bit 1 untuk i = 1, 2, …, m, dan blok cipherteks (C) adalah

$$C = (c_1, c_2, ..., c_m)$$

yang dalam hal ini c_i adalah bit 0 atau bit 1 untuk i = 1, 2, …, m.

Bila plainteks dibagi menjadi n buah blok, barisan blok-blok plainteks dinyatakan sebagai

$$(P_1, P_2, \ldots, P_n)$$

Untuk setiap blok plainteks P_i, bit-bit

penyusunnya dapat dinyatakan sebagai vektor

$$P_i = (p_{i1}, p_{i2}, \ldots, p_{im})$$

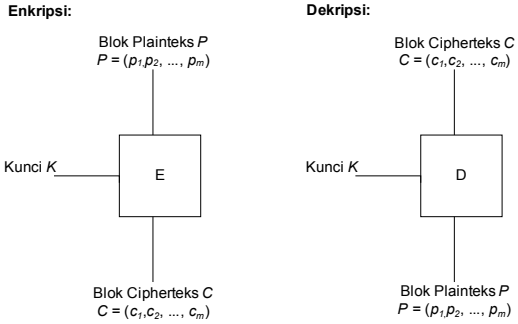
Enkripsi dengan kunci *K* dinyatakan dengan persamaan

$$E_k(P) = C,$$

sedangkan dekripsi dengan kunci *K* dinyatakan dengan persamaan

$$D_k(C) = P$$

Skema enkripsi dan dekripsi dengan *cipher* blok dapat dilihat pada gambar berikut.

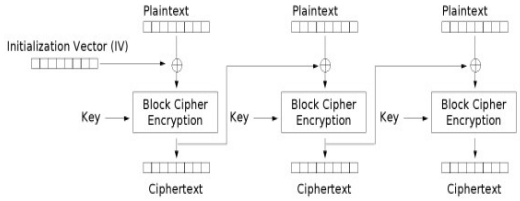


Gambar 6 Skema Enkripsi dan Dekripsi dengan *Cipher* Blok

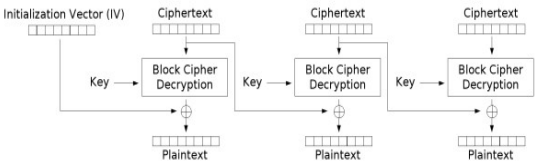
Pada kedua mode operasi ini digunakan IV (Initialization Vector), yaitu sebuah blok 'dummy' untuk melakukan proses enkripsi pada blok pertama, serta memberikan atribut acak pada proses tersebut. Pada umumnya IV tidak perlu dirahasiakan, namun sebaiknya IV tidak pernah digunakan pada enkripsi dengan kunci yang sama. Pada CBC, penggunaan ulang IV dapat memberikan informasi tentang blok pertama plainteks. Sedangkan pada CTR, penggunaan ulang IV akan menghancurkan keamanan mode operasi enkripsi tersebut. Pada mode CBC, IV harus dibuat secara acak pada awal proses enkripsi.

1. CBC (Cipher Block Chaining)

Mode operasi CBC ditemukan oleh IBM pada tahun 1976. Pada mode ini, tiap blok dari plainteks dilakukan XOR dengan hasil cipherteks dari blok sebelumnya yang kemudian dilakukan enkripsi. Dengan cara ini, tiap cipherteks dari masing-masing blok akan tergantung pada seluruh hasil cipherteks dari blok-blok sebelumnya. Selain itu, untuk membuat tiap pesan menjadi unik, digunakan IV (Initialization Vector) untuk dilakukan XOR dengan blok pertama.



Gambar 7 Enkripsi pada Mode CBC



Gambar 8 Dekripsi pada Mode CBC

Jika blok pertama memiliki indeks 1, maka rumus matematis untuk enkripsi pada mode CBC adalah :

$$C_i = E_K(P_i \oplus C_{i-1}), C_0 = IV$$

sedangkan rumus matematis untuk dekripsi pada mode CBC adalah :

$$P_i = D_K(C_i) \oplus C_{i-1}, C_0 = IV$$

keterangan :

C_i : Cipherteks pada blok i
P_i : Plainteks pada blok i
$E_k(...)$: Fungsi enkripsi yang digunakan
$D_k(...)$: Fungsi dekripsi yang digunakan
IV : Initialization Vector

Hingga saat ini, CBC merupakan mode operasi enkripsi cipher blok yang paling sering digunakan. Kelemahan utamanya adalah bahwa proses enkripsi pada CBC dilakukan secara sekuensial (sehingga tidak dapat dilakukan paralelisasi), dan bahwa harus dilakukan padding pada pesan sehingga berukuran sama dengan ukuran blok cipher. Salah satu cara untuk mengatasinya adalah dengan menggunakan metode ciphertext stealing. Metode ini tidak akan dijelaskan karena tidak digunakan dalam library PidCrypt.

Kita harus memperhatikan bahwa perubahan satu bit pada plainteks akan mengubah seluruh blok cipherteks selanjutnya. Sebuah plainteks dapat diselamatkan hanya dari dua blok cipherteks yang bersebelahan, sehingga dekripsi dapat dilakukan secara paralel, dan perubahan satu bit dari cipherteks dapat merusak hasil plainteks dari dekripsi dan mengubah bit yang bersesuaian pada blok

plainteks tersebut.

2. CTR (Counter)

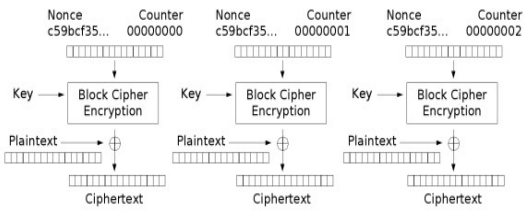
Mode counter sering disebut juga ICM (Integer Counter Mode) dan SIC (Segmented Integer Counter)

Seperti OFB, mode CTR mengubah cipher blok menjadi cipher stream. Mode ini membangkitkan blok keystream selanjutnya dengan mengenkripsi nilai berkelanjutan dari suatu 'counter'. Counter tersebut dapat berupa fungsi apapun yang mengeluarkan suatu sekuens yang dijamin tidak akan berulang dalam jangka waktu yang lama. Meskipun demikian, counter biasa (1, 2, 3, ... dst) lebih mudah dan sering digunakan.

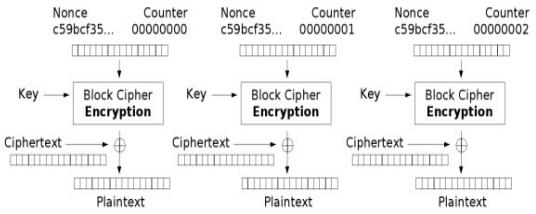
Penggunaan input yang sederhana dan dapat diprediksi memunculkan beberapa perdebatan kontroversial, yang menyebutkan bahwa "mengekspos suatu sistem kriptografi pada suatu input sistematis yang diketahui merupakan resiko yang tidak perlu". Hingga kini, mode CTR diakui secara luas, dan masalah yang disebabkan oleh fungsi input tersebut dianggap sebagai kelemahan dari cipher blok, bukan dari mode CTR. Meskipun demikian, terdapat serangan yang dikhususkan seperti Hardware Fault Attack yang menggunakan fungsi counter sederhana sebagai input.

CTR memiliki karakteristik menyerupai OFB, namun juga memperbolehkan penggunaan properti secara acak pada proses dekripsi. Mode CTR sangat cocok untuk beroperasi pada komputer multi-processor dimana blok dapat dienkripsikan secara paralel.

Perlu diperhatikan bahwa nonce pada gambar di bawah sama dengan IV pada CBC. IV dan counter dapat digabungkan, atau dilakukan XOR, sehingga menjadi blok counter unik yang sebenarnya untuk proses enkripsi.



Gambar 9 Dekripsi pada Mode CTR



Gambar 10 Dekripsi pada Mode CTR

E. Penggunaan Library PidCrypt

Bab ini akan membahas penggunaan fungsi enkripsi/dekripsi pada PidCrypt. Fungsi yang akan dijelaskan adalah fungsi enkripsi/dekripsi dengan algoritma AES dengan mode CBC dan CTR.

1. Algoritma AES-CBC

Algoritma AES dengan mode CBC di PidCrypt mendukung AES-128, 192, dan 256. Fungsi ini menggunakan metode yang sama untuk membuat kunci yang diperlukan dan IV sebagai OpenSSL. Selain itu, metode seperti OpenSSL juga digunakan untuk padding dan penyimpanan salt. Oleh karenanya, enkripsi dan dekripsi kompatibel dengan mode AES-nnn-CBC pada OpenSSL versi sekarang (v 0.9.8g)

Untuk melakukan enkripsi, kode yang digunakan adalah sebagai berikut :

```
//new instance
var aes = new pidCrypt.AES.CBC();
//initialization with
//plaintext(String), password(String)
//and options
aes.initEncrypt(plain, password,
{nBits: bits});
//encrypt the plaintext and returns the
//encrypted text
var crypted = aes.encrypt();
```

Kode tersebut akan :

- membangkitkan 8 byte secara acak (salt)
- melakukan hash password dan salt sebanyak 3 kali dengan MD5 untuk membangkitkan kunci dan IV
- meng-encode plainteks dengan UTF-8
- melakukan padding pada plainteks sehingga menjadi blok berukuran 16 byte
- mengenkripsi plainteks dengan mode AES-CBC
- menambahkan "Salted__" + salt di depan cipherteks
- men-encode cipherteks dengan base64

Sedangkan untuk melakukan dekripsi, digunakan kode sebagai berikut :

```
//new instance
var aes = new pidCrypt.AES.CBC();
```

```
//initialization with crypted
//text(base64 String), password(String)
//and options
aes.initDecrypt(crypted, password,
{nBits: bits});
//decrypt the crypted text and returns
//the plaintext
var plain = aes.decrypt();
```

Kode tersebut akan :
- men-decode cipherteks dari base64
- menghapus ”Salted__” + salt dari cipherteks
- melakukan hash password + salt sebanyak 3 kali dengan menggunakan MD5 untuk membangkitkan kunci dan IV
- mendekripsi dengan mode AES-CBC
- menghapus padding
- men-decode plainteks dari UTF-8

2. Algoritma AES-CTR

Algoritma AES dengan mode CTR di PidCrypt mendukung AES-128, 192, dan 256. Mode ini didasarkan pada implementasi mode CTR oleh Chriss Veness. Mode CTR sementara ini belum mendukung OpenSSL.

Untuk melakukan enkripsi, kode yang digunakan adalah sebagai berikut :

```
//new instance
var aes = new pidCrypt.AES.CTR();
//initialization with
//plaintext(String), password(String)
//and options
aes.initEncrypt(plain, password,
{nBits: bits});
//encrypt the plaintext and returns the
//encrypted text
var crypted = aes.encrypt();
```

Kode tersebut akan :
- membangkitkan 8 byte nonce secara acak
- meng-encode plainteks dengan UTF-8
- mengenkripsi dengan mode AES-CTR
- menambahkan nonce sebagai prefix dari cipherteks
- meng-encode cipherteks dengan base64

Sedangkan untuk melakukan dekripsi, digunakan kode sebagai berikut :

```
//new instance
var aes = new pidCrypt.AES.CTR();
//initialization with crypted
//text(base64 String), password(String)
//and options
aes.initDecrypt(crypted, password,
{nBits: bits});
//decrypt the crypted text and returns
//the plaintext
var plain = aes.decrypt();
```

Kode tersebut akan :
- men-decode cipherteks dari base64
- menghapus nonce dari cipherteks
- mendekripsi dengan mode AES-CTR
- men-decode plainteks dari UTF-8

F. Kasus Uji

Untuk menguji algoritma AES dengan mode CBC dan CTR pada library PidCrypt, digunakan plainteks berikut :

> ”IN 2009, EARTH HOUR IS BEING TAKEN TO THE NEXT LEVEL, with the goal of 1 billion people switching off their lights as part of a global vote. Unlike any election in history, it is not about what country you ▪re from, but instead, what planet you ▪ re from. VOTE EARTH is a global call to action for every individual, every business, and every community. A call to stand up and take control over the future of our planet. Over 74 countries and territories have pledged their support to VOTE EARTH during Earth Hour 2009, and this number is growing everyday.”

dan menggunakan kata kunci : ”kunci”.

Enkripsi dengan menggunakan mode AES-CBC menghasilkan cipherteks sebagai berikut :

> ”U2FsdGVkX1/89UPB2ixgGIlpk3IO3
> iGIzp1pjsu+2xCFaXQOmw0u5M4K2
> Bsd35ng3r7eHFirSkuF4v4hZIq0i1r6M
> voG85qvTChlFah6czUtlaphFIQMwJfJ
> vzg0e0F+TVJquGBDSE8o6AWyL621
> Hxrmy8x7QnW1b2xfZkKqRf7e+gSq
> Wbp4SodHgcxpkilB3GqvTDsY1zCdk
> 6ZsaiE+d2UhVdufMY1NyJggREgbQ
> U6mZcGbOwEdQfRZTamr3eyjGiHW
> X3f5Pa3QPTgKF7DZLoT1F3E+Kb/J
> US7yRsYvIjH8qwMPUj6A48NKCzsf
> SLQFaMHbmqAOF7GfNS/DC6Z0rzB
> CI7SLG0bLCYavjgdgVoyP4I7lk8oVX
> QOJIQJSJx0jgHHPUvNEuRR69OfyB
> stzPBR6GIF1zK/sKA+GUYcISWqS6
> YoY0ppsTk+IigiggZX3coSWxLMFrP
> vFiYY9T31/B14u5aPqjMQwYfS5mn
> DowcHugR+i9LWOY8JTuv31btaF6A
> qHbDxHL3TmFBDQC7kYR58W0OiP
> nPhAqUhaCh32XJuP9EF3f0UxDgfhF
> OvWcU7dAIE7Pd7zYfdHdwgUP30K
> KYc6cUBsU1PavPTC1FwTuSOs0Roo
> 5c4ioICQXelVkttMKh5Fy1z4BSOaqh
> 4HmGkcuz09qimvV4Qo+ge1LAVm/
> +OqbPPIN5hFK5wAy0izNvs8TcWgN
> JawrYz9QxOWXXuotXqBcjSRyL2bw
> PKWTkQX68XDXox0e5TBLkZGHut
> QVwq3”

Sedangkan enkripsi dengan menggunakan mode AES-CTR menghasilkan cipherteks sebagai berikut :

”sbbLScTExMQ9z/8EMx4np2x8yw6
KeD0DZU0p3974MUb09TQIvFuOaS
RiEkiq0/xP2yUBjxy9JCQD/Vwuw5yj
xll96ulkTmDZLwJ2nekV+EEX0TXm
64/MRlH+
+oV/rRtbyBZUOQKfWXhfBSjSEHt2
1wJ5VUjovxWSEhY2t5SMLCoPB/aH
zWdL1bAA31X+3P7vVaFm0D+bf0qn
ieC25aHFhKysxpOLtDP/A0T9fQsDK
v1nYKGq38bKjcjgLHCLO4lcQcaPhV
E6I4XsII4lXShnh4u2YQ9AamNOPdu
KXczE/ohzu+tp1N0Be4OaIe2f7IHasm
1fx57CpAVLIc4mByTSJ6P3tsKKwPK
hKqeHBrGTtZPgqMWLf8CiTQKMw
b1tiNcGaEnJQ96OYQXgqB1v+YP8yi
bQLNLbvRrVSMf3zQAB2bRQQlzxq
xCClY1p905DKB4G6rAoyAPNtVTP
LjElLm9En1D3IgziiCyR+hmZsn8neX
6ig7atEPjpMadFCtlRXz/DLrSB9rGBn
qqkV1zEbmw4WxOemfF2dhKjnIrYF
QKhCWQKlMxbcSENzBCgZtcj/CYE
Qy9/jv3PNIrs/H0y53TaxXw3kLOmz9
4T0fiekwqoscbiNmL3GdKDAx4nGhF
iRVt92j7FD6gz5iMRzqcIy7G9esNtET
ZgAFz/oV30BlcpklmWirK2fgZqf4um
nQIdzjBdJ5XG1GeYkdnshQmMZmB
WlwU21luyeLmMpDInLWd8dsmruH
7K”

Keduanya dapat didekripsi dengan baik dan menghasilkan plainteks yang sama dengan sebelum dilakukan enkripsi.

G. Kesimpulan

Kesimpulan yang dapat diambil dari keseluruhan makalah ini adalah :

- Penggunaan library PidCrypt dapat memudahkan pengembang aplikasi web untuk mengimplementasikan enkripsi di sisi klien.
- Library PidCrypt mendukung beberapa algoritma enkripsi, di antaranya adalah : MD5, SHA-1, SHA-256, AES-CBC, AES-CTR, RSA.
- Penggunaan algoritma AES dengan mode CBC maupun CTR menghasilkan cipherteks yang cukup rumit, sehingga sulit untuk dipecahkan.
- Fungsi AES-CBC dan AES-CTR yang diimplementasikan pada PidCrypt dapat berjalan dengan baik baik dalam proses enkripsi maupun dekripsi.

Daftar Pustaka

- Munir, Rinaldi. (2003). Diktat Kuliah Kriptografi..
- Slide Kuliah Kriptografi 2009.
- http://www.informatika.com/~rinaldi/ tanggal akses : 24 Maret 2009.
- http://www.pidder.com/pidcrypt/ tanggal akses : 24 Maret 2009
- http://www.sourceforge.net/ tanggal akses : 24 Maret 2009
- http://www.wikipedia.org/ tanggal akses : 24 Maret 2009