Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)
SAINTEKS 2020
ISBN: 978-602-52720-7-3
Februari 2020
Hal 420-425

Pengamanan File Teks Menggunakan Algoritma RC4

Bister Purba, Felisiana Apustriani Gulo, Nur Indah Utami, Yola Annisa Sihotang

Program Studi Teknik Informatika, STMIK Budi Darma, Medan Indonesia
Email: [1]felisianarni04@gmail.com, [1]nurindahu13@gmail.com, [1]yolaannisa31@yahoo.com

Abstrak−Kriptografi dibedakan menjadi 2 macam yaitu : Kriptografi kunci Asimetris dan Kriptografi kunci simetris. Algoritma RC4 adalah metode enkripsi terkenal dalam kriptografi.simetris. Disebut algoritma RC4 karena kunci yang digunakan untuk mengenkripsi sama dengan yang digunakan untuk mendeskripsi suatu data informasi *(File Txet).* Data informasi *(File Text)* yang ada harus diamankan untuk menjaga kerahasiaannya dari serangan *brute force.* Serangan *brute force* adalah serangan yang digunakan untuk mencoba mendeskripsi data yang dienkripsi. Serangan semacam itu dapat digunakan saat tidak mungkin mengambil keuntungan daripada kelemahan lain yang ada dalam sistem enkripsi membuat tugas lebih mudah.

Kata Kunci: Kriptograpi, , *Brute Force*, Algoritma RC4, *File Text*, *File Security*

1. PENDAHULUAN

Pada jaman sekarang, perkembangan ilmu teknologi sangat penting untuk menunjang kehidupan, terkhususnya untuk menjaga keamanan dan keaslian dari sebuah data yang kita miliki. Banyak pihak yang menggunakan ilmu teknologi sebagai sarana untuk mendapatkan keuntungannya sendiri dengan merugikan orang lain. Maka untuk mencegah hal tersebut terjadi kita dapat mengamankan data yang kita miliki dengan menggunakan Algoritma RC4 dari serangan yang dapat membahayakan data yang kita miliki seperti dari serangan *Brute Force.*

Beberapa ancaman yang diberikan adalah masalah keamanan, kerahasiaan, dan keaslian data. Untuk melindungi suatu data atau pesan agar tidak dibaca oleh pihak yang tidak berwenang serta mencegah pihak-pihak tersebut melakukan perubahan data, menghapus data serta menyisipkan sesuatu yang tidak pantas yang mengganggu keaslian dari data tersebut. Maka, untuk mengamankan data tersebut digunakanlah algoritma RC4. Algoritma RC4 sendiri adalah kriptografi kunci simetris karena menggunakan kunci yang sama untuk mengenkripsi ataupun mendeskripsi suatu data. Algoritma RC4 terbagi menjadi 2 macam, yaitu: inisalisasi state-array dan penghasilan kunci enkripsi serta pengenkripsiannya.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

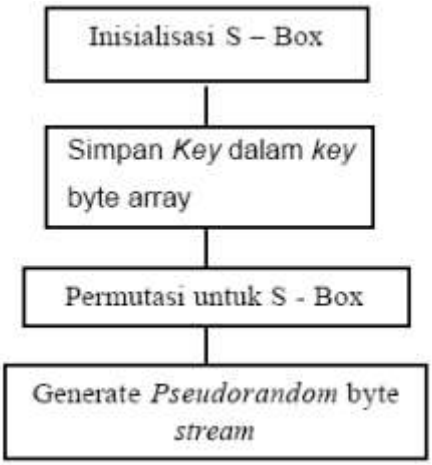
Kriptograpi adalah ilmu ataupun seni membuat sebuah pesan yang akan dikirim oleh pengirim dapat tersampaikan ke penerima dengan aman. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihal yang tidak sah. Perancangan algoritma kriptografi biasa disebut dengan kriptografer. [1] Kriptografi sebenarnya adalah mempelajari teknik matematis yang berhubungan dengan aspek keamanan dari suatu sistem informasi yaitu seperti kerahasiaan, integritas dari suatu data, autentikasi data, serta ketiadaan penyangkalan. Keempat aspek diatas merupakan tujuan dari fundamental pada sistem kriptografi.

a. Kerahasiaan *(Confidentiality)*
   Kerahasiaan adalah layanan yang biasa digunakan untuk menjaga suatu informasi dari setiap pihak yang tidak bersangkutan untuk mengaksesnya. Informasi ini hanya dapat di akses oleh pihak yang bersangkutan. Contoh serangannya adalah *sniffing*. Proteksi yang bisa dilakukan dengan metode enkripsi.
b. Integritas Data *(Data Integrity)*
   Integritas data merupakan layanan yang di tuju untuk mencegah terjadinya pengubahan informasi oleh pihak yang tidak bertanggung jawab. Integritas data harus di pastikan keasliannya agar sistem informasi yang ada mampu mendeteksi terjadinya manipulasi data. Manipulasi data di sini seperti penyisipan data, penghapusan data dan penggantian data. Contoh serangannya yaitu *spoofing*, virus, *trojan horse* atau bisa juga *man in the middle attack.* Sedangkan proteksi yang bisa dilakukan adalah dengan cara *signature, certificate* dan *hash.*
c. Autentikasi *(Authentication)*
   Autentikasi merupakan layanan yang terkait dengan identifikasi terhadap pihak-pihak yang ingin mengakses sistem informasi *(authentication)* ataupun keaslian data dari sistem informasi itu sendiri *(data origin authentication)*. Contoh serangannya adalah *password* palsu, terminal palsu, atau situs web palsu. Sedangkan untuk proteksinya dapat dilakukan dengan cara *certificates*.
d. Ketiadaan Penyangkalan *(nonrepudiation)*
   Ketiadaan penyangkalan merupakan layanan yang berfungsi untuk mencegah terjadinya penyangkalan terhadap suatu aksi yang dilakukan oleh pelaku sistem informasi.

2.2 Algoritma RC4

RC4 merupakan cipher aliran yang biasa digunakan secara luas untuk sistem keamanan, missal: protokol *Secure Socket Layer* (SSL). Algoritma kriptografi ini sederhana serta mudah diimplementasikan. RC4 diciptakan oleh Ron Rivest di laboratorium RSA. (RC adalah singkatan dari *Ron's Code*) [2]. Gambar 1.1 memperlihatkan rangkaian proses yang dijalankan untuk mengenkripsi atau mendeskripsi data.

ISBN: 978-602-52720-7-3
Februari 2020
Hal 420-425



Gambar 1. Rangkaian Proses RC4

RC4 membangkitkan *keystream* yang kemudian di XOR-kan dengan *plaintext* pada waktu enkripsi (atau di XOR kan dengan bit ciphertext pada waktu deskripsi). RC4 bukan seperti *cipher* aliran biasa yang memproses data yang terdapat dalam bit. RC4 memproses data yang ada dalam waktu hanya ukuran *byte* (1 *byte* = 8 bit). RC4 menggunakan 2 kotak substitusi atau (*S-box*) dengan *array* 256 *byte* dan yang berisi permutasi daripada bilangan 0 sampai dengan 255 dan *S-box* kedua berisi permutasi fungsi daripada kunci sepanjang variable. Akan tetapi, algoritma RC4 memiliki kelemahan yaitu : kenungkinan adanya nilai-nilai di dalam *array S* yang bernilai sama. Hal ini karena karakter kunci dikopi secara berulang. RC4 juga mudah diserang dengan *known-plaintext attack* jika kriptanalis mengetahui beberapa buah plaintext dan ciphertext yang berkorespondensi. Untuk mengatasi permasalahan tersebut ada beberapa cara yang dilakukan yaitu: [3]

a. Menggunakan kunci yang panjang (minimal panjang kunci kurang lebih 3 karakter dan maksimal lebih kurang dari 255 karakter) agar kemungkinan kunci yang dimasukkan secara berulang dalam *array* kunci semakin kecil dan memakai kombinasi yang berlainan .
b. Untuk mengenkripsi file yang berbeda, usahakan tidak menggunakan kunci yang sama.
c. Jika menggunakan kunci yang sama untuk setiap kali mengenkripsi file, kita memerlukan *initialization vector* (IV) pada *secret key*. Jika IV yang digunakan untuk setiap kali proses enkripsi tidak pernah sama, akan dihasilkan ciphertext yang berbeda meskipun plaintext yang dienkrisi sama.
d. Mengacak (mengubah susunan) plaintext sebelum diubah kedalam ciphertext, sehingga jika seorang pengganggu memperoleh 1 byte data dari plaintext, seorang pengganggu tersebut tidak dapat memperoleh data lainnya dengan cara meng-XOR-kan 2 buah ciphertex dan byte data yang diketahui.
e. Mengubah meode pengisian kunci kedalam kunci *array*. Caranya, pengisian kunci ke dalam *array* cukup sekali saja, kemudian sisa variable kunci *array* yang lainnya akan diisi dengan nilai yang dibangkitkan secara acak.

2.3 Serangan

Serangan adalah setiap usaha atau percobaan yang dilakukan oleh user untuk menemukan kunci atau menemukan plaintext dari ciphertext-nya. Jenis- jenis serangan bisa dibedakan berdasarkan keterlibatan penyerang dalam komunikasi, teknik yang digunakan untuk menemukan kunci,

1. Berdasarkan keterlibatkan penyerang dalam komunikasi

a. Serangan Pasif *(Passive Attack)*
Untuk serangan jenis ini, penyerang tidak terlibat dalam komunikasi antara pengirim dan penerima, hanya melakukan penyadapan untuk memperoleh data atau informasi sebanyak-banyaknya yang digunakan untuk kriptanalisis. Beberapa metode penyadapan data yaitu:
1) *Wiretapping*
Penyadap mencegat data yang ditransmisikan pada saluran kabel komunikasidengan menggunakan sambungan perangkat keras.
2) *Electromagnetic eavesdropping*
Penyadap mencegat data yang ditransmisikan melalui saluran nirkabel (wireless), seperti radio dan microwave.
3) *Acoustic eavesdropping*
Menangkap gelombang suatu yang dihasilkan oleh suara manusia.

b. Serangan Aktif *(Active Attack)*
Untuk serangan jenis ini penyerang mengintervensi komunikasi dan ikut mempengaruhi sistem untuk keuntungan dirinya serta mengubah aliran pesan seperti menghapus sebagian ciphertext, mengubah ciphertext, menyisipkan potongan ciphertext palsu, mengulang pesan lama, serta mengubah informasi yang tersimpan. Contoh serangan aktif adalah *man-in-the-middle attack.*

2. Berdasarkan teknik yang digunakan untuk menemukan kunci

a. *Exhaustive attack/brute force attack*

ISBN: 978-602-52720-7-3
Februari 2020
Hal 420-425

Brute Force adalah suatu pendekatan langsung *(straightforward)* yang digunakan dalam memecahkan suatu masalah. Hal ini biasanya didasarkan pada suatu pernyataan masalah *(problem statement)* serta definisi konsep yang terlibat. [4]

1) Mengungkap plaintext/kunci dengan mencoba semua kemungkinan kunci.
2) User harus menggunakan kunci yang lebih panjang dan tidak mudah ditebak.semakin lama waktu *exhaustive search* nya.

b. *Analytical attack*

Pada jenis serangan ini kriptanalis tidak mencoba semua kemungkinan kunci, tetapi menganalisis kelemahan algoritma kriptografi untuk mengurangi kemungkinan kunci yang tidak mungkin ada. Teknik analisis yang digunakan yaitu:

1) Linear Cryptanalysis
2) Differential Cryptanalysis
3) Integral Cryptanalysis

Untuk menghadapi serangan jenis ini, kriptografi harus membuat algoritma kriptografi yang kompleks. Secara umum metode *analytical attack* lebih cepat menemukan kunci dibandingkan dengan *exhaustive attack.*

3. Berdasarkan banyaknya informasi yang diketahui kriptanalis

a. Ciphertext-only attack

Kriptanalis memiliki ciphertext dari sejumlah pesan yang seluruhnya telah dienkripsi dengan algoritma yang sama yang digunakan untuk menemukan kunci untuk deskripsi. Kriptanalis menggunakan teknik exhaustive key search untuk mengubah plaintext menjadi ciphertext atau teknik analisis frekuensi untuk menerka informasi yang diketahui.
Diberikan : $C_1 = E_k(P_1)$, $C_2 = E_k(P_2)$,…, $C_i = E_k(P_i)$
Deduksi : $P_1, P_2, \ldots, P_i$ atau k untuk mendapatkan P_{i+1} dari $C_{i+1} = E_k(P_{i+1})$

b. Known-plaintext attack

Kriptanalisis memiliki akses tidak hanya ke ciphertext sejumlah pesan, namun juga mempunyai plaintext pesan tersebut. Serangan ini disebut pula cleartext attack. Secara formal formulasi known-plaintext attack adalah
Diberikan : $P_1, C_1, = E_k(P_1)$, P_2, $C_2 = E_k(P_2)$, …, P_i, $C_i = E_k(P_i)$
Deduksi : k untuk mendapatkan P_i+1 dari $C_i+1 = E_k(P_i+1)$

c. Chosen-plaintext attack

Kriptanalis tidak hanya memiliki akses atas ciphertext dan plaintext untuk beberapa pesan, tetapi juga dapat memilih penggalan dari plaintext yang dienkripsi.
Diberikan : $P_1, C_1 = E_k(P_1)$, $P_2, C_2 = E_k(P_2)$, …, P_1, $C_1 = E_k(P_i)$
Kriptanalis dapat memilih antara $P_1, P_2, \ldots, P_i$
Deduksi : k untuk mendapatkan P_{i+1} dari $C_{i+1} = E_k(P_{i+1})$

d. Adaptive-chosen-plaintext attack

Kriptanalis dapat memilih blok plaintext yang lebih kecil dan kemudian memilih yang lain berdasarkan hasil yang pertama. Proses tersebut dapat dilakukan secara terus menerus sampai kriptanalis mendapatkan semua informasi yang diinginkan.

e. Chosen-ciphertext attack

Kriptanalis memilih sejumlah ciphertext untuk dideskripsikan dan memiliki akses ke plaintext hasil deskripsi. Berdasarkan hasil deskripsinya, kriptanalis akan memilih ciphertext seperti berikut :
Diberikan : C_1, $P_1 = D_k(C_1)$, C_2, P_2, $= D_k(P_2)$, …, C_i, $P_i = D_k(C_i)$
Deduksi : k (mungkin diperlukan untuk mendeskripsi pesan pada waktu yang akan datang).

f. Chosen-key attack

Serangan ini merupakan kombinasi dari *chosen-plaintext attack* dan *Chosen-ciphertext attack.* Kriptanalis memiliki pengetahuan mengenai hubungan antara kunci yang berbeda dan memilih kunci yang tepat untuk mendeskripsi pesan.

4. Berdasarkan cara dan posisi seseorang untuk mendapatkan pesan dalam saluran komunikasi

a. Sniffing

Sniffing adalah pesan dalam suatu saluran komunikasi. Akibat dari kegiatan *Sniffing* yaitu :

1) Hilangnya privasi
2) Tercurinya informasi yang penting dan rahasia

Untuk mencegah serangan ini, lakukanlah enkripsi (SSL, SSH, atau PGP) dan tidak melakukan aktivitas yang sifatnya rahasia pada suatu jaringan computer yang belum dikenal seperti warnet atau kantor yang memiliki computer banyak dan dihubungkan dalam satu jaringan.

b. Replay attack

Replay attack merupakan serangan jaringan dimana penyerang menyadap percakapan antara pegirim dan penerima serta mengambil informasi autentik dengan berbagi kunci. Penyerang kemudian menghubungi penerima dengan kunci itu sebagai bukti identitas dan keaslian untuk menipu penerima.

c. Spoofing

Spoofing merupakan teknik untuk memperoleh akses yang tidak sah ke suatu computer atau informasi dimana penyerang berhubungan dengan pengguna dengan berpura-puramenjadi *host* yang dapat dipercaya. *Spoofing* biasanya dilakukan oleh seorang *hacker/cracker*. Ada beberapa *Spoofing* yaitu:

1) IP
2) MAC address,

ISBN: 978-602-52720-7-3
Februari 2020
Hal 420-425

3) DNS
4) Routing

Untuk mencegah nya, bisa menggunakan beberapa cara :
1) Mengimplementasikan firewall dengan benar
2) Memakai *patch* yang mencegah prediksi *sequence number*
3) Mengeset *router* agar tidak bisa dilewati kecuali melalui rute yang telah ditentukan.

d. Man-in-the-middle [1]

Jika *Spoofing* hanya menipu satu pihak, *Man-in-the-middle* dapat menipu lebih dari satu pihak.

3. ANALISA DAN PEMBAHASAN

Algoritma RC4 menggunakan mode 4byte untuk mengenkripsikan plaintext **NURI dengan kunci 13**

1. Lakukan inisialisai array S-box dengan panjang 4 byte. S[0]=0, S[1]=1, S[2]=2, S[3]=3.

INDEX	0	1	2	3
S	0	1	2	3

2. Inisialisasikan *array* kunci S-box yang lain. Kunci K diulang dengan panjang 4 byte. K[0]=0, K[1]=1, K[2]=2, K[3]=3.

INDEX	0	1	2	3
K	0	1	2	3

3. Lakukan permutasi terhadap nilai-nilai di dalam array S dengan cara menukarkan isi array S[i] dengan S[j] dengan mode 4 prosesnya berikut:
   j = 0
   For i = 0 to 3
   j = (j + S[i] + K[i]) mod 4
   isi S[i] dan isi S[j] ditukar.
   Dengan menggunakan algoritma tersebut, untuk nilai i = 0 sampai i = 3 didapatkan nilai array S berikut:
   1) Iterasi pertama, untuk nilai i = 0
      j = (j + S[i] + K[i]) mod 4
      j = (j + S[0] + K[0]) mod 4
      j = (0 + 0 + 1) mod 4
      j = 1
      Dilakukan penukaran isi array S[0] dan S[1] sehingga array S berbentuk:

Index	0	1	2	3
S	1	0	2	3

   2) Iterasi kedua untuk nilai j = 1 dan i = 1
      j = (j + S[i] + K[i]) mod 4
      j = (1 + S[1] + K[1]) mod 4
      j = (1 + 0 + 3) mod 4
      j = 0
      Dilakukan penukaran isi array S[1] dan S[0] sehingga array S berbentuk:

Index	0	1	2	3
S	0	1	2	3

   3) Iterasi ketiga untuk nilai j = 0 dan i = 2
      j = (j + S[i] + K[i]) mod 4
      j = (0 + S[2] + K[2]) mod 4
      j = (0 + 2 + 1) mod 4
      j = 3
      Dilakukan penukaran isi array S[2] dan S[3] sehingga array S berbentuk:

Index	0	1	2	3
S	0	1	3	2

   4) Iterasi keempat untuk nilai j = 3 dan i = 3
      j = (j + S[i] + K[i]) mod 4
      j = (3 + S[3] + K[3]) mod 4
      j = (3 + 2 + 3) mod 4
      j = 0
      Dilakukan penukaran isi array S[3] dan [0] sehingga array S berbentuk:

Index	0	1	2	3
S	2	1	2	0

ISBN: 978-602-52720-7-3
Februari 2020
Hal 420-425

4. Untuk mendapatkan *ciphertext*, terlebih dahulu bangkitkan keystream sebanyak 3 byte, proses membangkitkan kunci enkripsi pada mode 4byte sebagai berikut:
   i = j = 0
   i = (i + 1) mod 4
   i = (j + S[i]) mod 4
   isi S[i] dan S[j] ditukar
   t = (S[i] + S[j]) mod 4
   K = S[t]
   Dengan menggunakan algoritma tersebut, bisa membangkitkan 4 buah kunci proses enkripsi sebagai berikut:
   1) Iterasi pertama, untuk nilai i = j = 0, maka
      i = (i + 1) mod 4
      i = (0 + 1) mod 4
      i = 1
      j = (j + S[i]) mod 4
      j = (0 + S[1]) mod 4
      j = (0 + 1) mod 4
      j = 1
      Menukarkan isi S[1] dan S[1] sehingga array S berbentuk:

Index	0	1	2	3
S	2	1	3	0

      t = (S[1] + S[1]) mod 4
      t = (1 + 1) mod 4
      t = 2
      K = S[t] = S[2] = 3
      Jadi, kunci pertama untuk enkripsi adalah = 3.
   2) Iterasi kedua, untuk nilai i = 1 dan j = 1 maka
      i = (i + 1) mod 4
      i = (1 + 1) mod 4
      i = 2
      j = (j + S[i]) mod 4
      j = (1 + S[2]) mod 4
      j = (1 + 3) mod 4
      j = 0
      Menukarkan isi S[2] dan S[0] sehingga array S berbentuk:

Index	0	1	2	3
S	3	1	2	0

      t = (S[2] + S[0]) mod 4
      t = (2+ 3) mod 4
      t = 1
      K = S[t] = S[1] = 1
      Jadi, kunci kedua untuk enkripsi adalah = 1.
   3) Iterasi ketiga, untuk nilai i = 2 dan j = 0 maka
      i = (i + 1) mod 4
      i = (2 + 1) mod 4
      i = 3
      j = (j + S[i]) mod 4
      j = (0 + S[3]) mod 4
      j = (0 + 0) mod 4
      j = 0
      Menukarkan isi S[3] dan S[0] sehingga array S berbentuk:

Index	0	1	2	3
S	0	1	2	3

      t = (S[3] + S[0]) mod 4
      t = (3+ 0) mod 4
      t = 3
      K = S[t] = S[3] = 3
      Jadi, kunci ketiga untuk enkripsi adalah = 3.
   4) Iterasi keempat, untuk nilai i = 3 dan j = 0 maka
      i = (i + 1) mod 4
      i = (3 + 1) mod 4

ISBN: 978-602-52720-7-3
Februari 2020
Hal 420-425

i = 0
j = (j + S[i]) mod 4
j = (0 + S[0]) mod 4
j = (0 + 0) mod 4
j = 0
Menukarkan isi S[0] dan S[0] sehingga array S berbentuk:

Index	**0**	**1**	**2**	**3**
S	0	1	2	3

t = (S[0] + S[0]) mod 4
t = (0+ 0) mod 4
t = 0
K = S[t] = S[0] = 0
Jadi, kunci keempat untuk enkripsi adalah = 0.

5. Proses enkripsinya *plaintext* di XOR-kan dengan kunci.
Untuk plaintext **NURI** dan kunci 3130 dari hasil pembangkitan kunci maka di XOR-kan dalam bentuk kode biner 8 byte.

Plaintext				
Desimal	78	85	82	73
Symbol	N	U	R	I
Biner	01001110	01010101	01010010	01001001

Keystream				
Desimal	3	1	3	0
Symbol	ETX	SOH	ETX	NUL
Biner	00000011	00000001	00000011	00000000

K ⊕ P (Biner)				
Keystream	00000011	00000001	00000011	00000000
Plaintext	01001110	01010101	01010010	01001001
K ⊕ P	01001101	01010100	01010001	01001001
Ciphertext (Simbol)	**M**	**T**	**Q**	**I**
Ciphertext (Desimal)	**77**	**84**	**81**	**73**

Maka didapat ciphertext hasil enkripsi implementasi algoritma RC4 mengunakan mode 4byte yaitu **MTQI.**
Cara mendapatkan plaintext:

K ⊕ P (Biner)				
Keystream	00000011	00000001	00000011	00000000
Ciphertext	01001101	01010100	01010001	01001001
K ⊕ P	01001110	01010101	01010001	01001001
Ciphertext (Simbol)	**N**	**U**	**R**	**I**
Ciphertext (Desimal)	**78**	**85**	**82**	**73**

Maka didapat plaintext hasil deskripsi implementasi algoritma RC4 menggunakan mode 4byte yaitu **NURI.**

4. KESIMPULAN

Penggunaan Algoritma RC4 dapat kita digunakan untuk memberi pengamanan pada data teks atau data informasi apapun yang kita miliki, sehingga dengan menggunakan algoritma kriptografi RC4 kita dapat mengamankan semua data yang kita miliki dari serangan *brute force*.

REFERENCES

[1] S. Bruce, Applied Cryptography : Protocols, Algorithms, and Source Code in C, New York: Jhon Wiley & Sons, 1996.
[2] W. H. Haji and S. Mulyani, "Implementasi RC4 Stream Cipher untuk Keamanan Data Dasar," Seminar Nasional Aplikasi Teknologi Informasi, 2012.
[3] A. Irfianti, "Metode Pengamanan Enkripsi RC4 Stream Cipher untuk Aplikasi Pelayanan Gangguan," Seminar Nasional Aplikasi Teknologi Informasi 2007, pp. C49-C52, 2007.
[4] B. W. Santoso, F. Sundawa and M. Azhari, "Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Enginer) Berbasis WEB Pada Database," Jurnal Sisfotek Global, vol. 6, p. 2, 2016.
[5] R. Sadikin, Kriptografi Untuk Keamanan Jaringan, Yogyakarta: Andi Offset, 2012.
[6] R. Munir, Kriptografi, Bandung: Informatika Bandung, 2006.