STUDI ALGORITMA SOLITAIRE CIPHER

Puthut Prabancono – NIM : 13506068

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16068@students.if.itb.ac.id

Abstrak

Penggunaan kriptografi untuk merahasiakan pesan dapat dilakukan dengan berbagai macam cara. Algoritma Solitaire Cipher menggunakan kartu sebagai alat bantu utama. Sebuah set kartu biasanya terdiri dari 52 kartu ditambah dua atau tiga buah kartu joker. Jumlah ini sangat pas untuk digunakan untuk melakukan operasi-operasi terhadap sebuah teks mengingat angka 52 merupakan kelipatan dari jumlah huruf dalam alfabet (26), sehingga masing-masing kartu dapat merepresentasikan sebuah huruf. Algoritma ini sebelumnya dikenal dengan nama Pontifex, untuk menyembunyikan fakta bahwa algoritma ini menggunakan kartu sebagai alat bantu utama.

Penggunaan satu set kartu untuk merahasiakan pesan merupakan salah satu cara yang mudah digunakan. Cara ini juga murah untuk diterapkan karena tidak membutuhkan komputer atau mesin lain untuk menggunakannya.

Makalah ini membahas tentang cara kerja Solitaire Cipher secara keseluruhan. Termasuk di dalamnya bagaimana teknik untuk memanfaatkan jumlah kartu untuk menentukan substitusi dan cara melakukan enkripsi beserta dekripsinya.

Algoritma kriptografi lain yang memiliki sedikit kemiripan dan digunakan sebagai pembanding dengan Solitaire Cipher adalah Vigenere Cipher. Keduanya melakukan substitusi pada plaintext dengan menggunakan kunci sehingga beberapa huruf yang sama pada plaintext dapat berubah menjadi huruf-huruf yang berbeda pada ciphertext. Perbedaannya adalah cara untuk menentukan kunci dan juga cara menggunakan kunci untuk menghasilkan ciphertext.

Implementasi Solitaire Cipher bukan berupa program yang sudah jadi , tetapi hanya potongan source code. Selain itu juga akan dibahas mengenai feasibilitas reliabilitas algoritma ini untuk diterapkan dan juga contoh pengenkripsian plainteks pendek yang kemudian didekripsi kembali.

***Kata kunci**: Solitaire cipher*, *vigenere cipher*, *one-time pad*, kunci aliran, kriptografi, enkripsi, dekripsi.

1. Pendahuluan

1.1. Kriptografi

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Kriptografimodern menggunakan gagasan dasar yang sama seperti kriptografi klasik, yaitu permutasi dan transposisi, tetapi penekanannya berbeda.

Pada kriptografi klasik, kriptografer menggunakan algoritma yang sederhana, yangmemungkinkan cipherteks dapat dipecahkan dengan mudah (melalui penggunaanstatistik, terkaan, intuisi, dan sebagainya). Sedangkan algoritma kriptografi modern dibuatsedemikian rupa sehingga kriptanalis sangat sulit memecahkan cipherteks tanpa mengetahui kunci.

Algoritma kriptografi modern umumnya beroperasi dalam mode bit ketimbang modekarakter. Operasi dalam mode bit berarti semua data dan informasi dalam bentukrangkaian bit. Rangkaian bit yang menyatakan plainteks dienkripsi menjadi cipherteksdalam bentuk rangkaian bit, demikian sebaliknya. Perkembangan algoritma kriptografi modern berbasis bit didorong oleh penggunaan komputer digital yang merepresentasikan data dalam bentuk biner.

Algoritma kriptografi yang beroperasi dalam mode bit dapat dikelompokkan mejadi duakategori:

a. cipher aliran (stream cipher)
Algoritma kriptografi beroperasi pada plainteks/ cipherteks dalam bentuk bit tunggal, yangdalam hal ini rangkaian bit dienkripsikan/didekripsikan bit per bit.

b. cipher blok (blok cipher)

Algoritma kriptografi cipher blok beroperasi pada plainteks/cipherteks dalam bentuk blokbit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudahditentukan sebelumnya. Misalnya panjang blok adalah 64 bit, maka berarti algoritma enkripsi memerlakukan 8 karakter setiap kali penyandian (1 karakter = 8 bit dalampengkodean ASCII)
Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci (yang ukurannya sama dengan ukuran blok plainteks). Algoritma enkripsi menghasilkan blok cipherteksyang berukuran sama dengan blok plain teks.

Algoritma blok cipher menggabungkan beberapa teknik kriptografi klasik dalam prosesenkripsi. Dengan kata lain, cipher blok dapat diacu sebagai super-enkripsi. Teknik-teknikkriptografi klasik yang digunakan adalah:

1.Substitusi. Teknik ini mengganti satu atau sekumpulan bit pada blok plainteks tanpamengubah urutannya.

2.Transposisi atau permutasi. Teknik ini memindahkan posisi bit pada blok plainteksberdasarkan aturan tertentu.

Selain itu, blok cipher juga menggunakan dua teknik tambahan, yaitu:

3. Ekspansi. Teknik ini memerbanyak jumlah bit pada blok plainteks berdasarkan aturan tertentu, misalnya dari 32 bit menjadi 48 bit. Dalam praktek, aturan ekspansi dinyatakan dengan tabel.

4. Kompresi. Teknik ini kebalikan dari ekspansi, di mana jumlah bit pada plainteks diciutkan berdasarkan aturan tertentu. Dalam praktek, aturan ekspansi dinyatakan dengan tabel.

Pada tahun 1949, Shannon mengemukakan dua prinsip (properties) penyandian(encoding) data. Kedua prinsip ini dipakai dalam perancangan cipher blok yang kuat. Kedua prinsip Shannon tersebut adalah confusion dan diffusion. Prinsip confusion adalah menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci. Prinsip diffusion menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyakmungkin cipherteks. Sebagai contoh, pengubahan kecil pada plainteks sebanyak satuatau dua bit menghasilkan perubahan pada cipherteks yang tidak dapat diprediksi. Untukmendapatkan keamanan yang bagus, prinsip confusion dan diffusion diulang berkali-kalipada sebuah blok tunggal dengan kombinasi yang berbeda-beda.

1.2. Algoritma Solitaire Cipher
Algoritma Solitaire Cipher salah satu algoritma kriptografi klasik. Algoritma ini dikembangkan oleh Bruce Schneier, seorang pakar keamanan yang juga pernah membuat algoritma Blowfish pada tahun 1999. Solitaire Cipher termasuk algoritma berbasis substitusi, yaitu algoritma yang mengganti satu atau beberapa karakter menjadi satu atau beberapa karakter lain.

Pada awalnya algoritma Solitaire Cipher ini diciptakan untuk digunakan dalam sebuah novel berjudul Cryptonomicon karangan Neal Stephenson. Dalam novel tersebut, Solitaire Cipher ini digunakan oleh agen rahasia Soviet yang sangat berisiko jika mengirim pesan menggunakan alat bantu lain seperti komputer. Menurut Bruce, algoritma ini sangat mudah digunakan karena hanya membutuhkan satu set kartu dan seperangkat alat tulis.

Pada novel Cryptonomicon, algoritma ini tidak dikenal dengan nama Solitaire Cipher, melainkan Pontifex. Hal ini dilakukan untuk menyembunyikan kenyataan bahwa algoritma ini menggunakan alat bantu berupa kartu.

Algortima Solitaire Cipher ini sendiri bukan merupakan sebuah algoritma untuk mengenkripsi apapun, tetapi algoritma ini menghasilkan sebuah string dari nilai-nilai semi acak yang kemudian akan digunakan untuk proses enkripsi yang sebenarnya.

String dari nilai-nilai ini dihasilkan dari kunci yang diberikan oleh pengguna. Kunci yang dimaksud dalam hal ini adalah urutan kartu-kartu saat inisialisasi. Nilai keluaran "acak" akan selalu sama untuk setiap kunci. Karena nilai keluaran ini digunakan untuk mengenkripsi, maka kita dapat menganggap kunci ini merupakan password untuk proses enkripsi dan dekripsi, dan apabila kita tidak mengetahui kunci

tersebut,maka tidak akan ada harapan untuk mendapatkan nilai-nilai yang untuk melakukan dekripsi pesan.

Algoritma ini mungkin terlihat *low-tech* tetapi sebenarnya ditujukan untuk menjadi algoritma yang *hi-tech*. Solitaire Cipher didesain supaya aman bahkan terhadap agen militer dengan komputer tercanggih dan kriptanalis terpandai.

Algoritma ini cukup lambat, sehingga membutuhkan waktu yang lama untuk melakukan enkripsi dan dekripsi pesan yang sangat panjang. Oleh karena itu, usaha untuk menebak-nebak kunci untuk algoritma ini akan memerlukan waktu yang sangat lama.

Dalam penggunaannya, pengirim dan penerima pesan harus memiliki satu set kartu dengan urutan yang sama persis agar pesan yang terenkripsi oleh pengirim dapat dienkripsi dengan benar oleh pihak penerima.

2. Cara Kerja

Sebelum membahas lebih lanjut mengenai algoritma yang digunakan untuk menghasilkan kunci untuk enkripsi dan dekripsi, kita akan menyamakan persepsi mengenai set kartu yang digunakan.

Satu set kartu berisi 54 kartu dengan komposisi dua buah joker dan empat set gambar dengan masing-masing set terdiri dari 13 kartu.

Untuk menyederhanakan pembahasan dalam makalah ini, hanya akan digunakan dua buah joker dan dua set gambar kartu. Jumlah ini sudah cukup untuk menerapkan algoritma solitaire cipher karena sudah terdiri dari dua joker yang digunakan sebagai pembatas, dan 26 kartu lain yang masing-masing diberi nilai satu sampai dua puluh enam .

Kedua joker diberi nilai masing-masing 27 dan 28. Dengan demikian, pada set kartu yang digunakan terdiri dari kartu-kartu dengan nilai 1 sampai 28.

2.1. Algoritma

Algoritma untuk menghasilkan kunci untuk proses enkripsi dan dekripsi terdiri dari enam langkah. Enam tahap ini akan menghasilkan sebuah angka yang merupakan salah satu bagian aliran kunci. Enam tahap ini kemudian dilakukan kembali untuk mendapatkan kunci kedua dan terus diulang sampai panjang kunci yang diinginkan atau disepakati.
Keenam langkah tersebut adalah sebagai berikut:

1. Urutkan tumpukan kartu berdasarkan kunci tertentu. Bagian ini adalah bagian paling penting, karena pihak yang mengetahui sebuah nilai awal dari deck dapat dengan mudah mendapatkan nilai yang sama darinya. Bagaimana sebuah tumpukan diinisialisasi terserah oleh penerima. Mengocok kartu dengan benar-benar acak akan lebih baik, walaupun masih ada beberapa metode lain.

Misal urutan awal seperti ini:
1 4 7 10 13 16 19 22 25 **28** 3 6 9 12 15 18 21 24 **27** 2 5 8 11 14 17 20 23 26

2. Temukan joker A (yang bernilai 27). Pindahkan satu kartu ke bawah (dengan kata lain, menukar dengan satu kartu dibawahnya). Jika joker tersebut berada di tumpukan paling bawah, pindahkan ke tepat di bawah tumpukan teratas.

Urutan kartu tersebut akan menjadi seperti ini:
1 4 7 10 13 16 19 22 25 **28** 3 6 9 12 15 18 21 24 2 **27** 5 8 11 14 17 20 23 26

3. Temukan joker B (yang bernilai 28). Pindahkan dua kartu ke bawah. Jika joker tersebut berada pada tumpukan terbawah, pindahkan ke bawah kartu kedua dari atas (jadi kartu ketiga). Jika joker tersebut berada pada posisi kedua dari bawah, pindahkan ke bawah kartu teratas (menjadi kartu kedua).

Setelah langkah ketiga ini, urutan kartu akan menjadi:
1 4 7 10 13 16 19 22 25 3 6 **28** 9 12 15 18 21 24 2 **27** 5 8 11 14 17 20 23 26

Langkah pertama dan kedua harus dilakukan secara berurutan. Jika ragu-ragu, ingat-ingat untuk memindahkan joker A sebelum joker B. Hati-hati juga bila joker A dan B berada pada tumpukan terbawah. Jika joker adalah kartu terbawah, anggap kartu itu kartu pertama, sebelum memulai berhitung.

4. Lakukan *triple cut*. Yaitu ganti kartu-kartu yang berada di bagian kiri kartu joker pertama dengan kartu-kartu di bagian kanan dari kartu joker kedua. Perlu diperhatikan bahwa joker pertama adalah joker yang berada di posisi lebih tinggi dari kartu joker lain, tidak penting apakah itu joker A atau B.

Susunan kartu sebelum *triple cut* dilakukan:
1 4 7 10 13 16 19 22 25 3 6 **28** 9 12 15 18 21 24 2 **27** 5 8 11 14 17 20 23 26

Susunan kartu setelah *triple cut* dilakukan:
5 8 11 14 17 20 23 26 **28** 9 12 15 18 21 24 2 **27** 1 4 7 10 13 16 19 22 25 3 6

Perlu diperhatikan juga bahwa posisi kartu-kartu yang berada di antara kedua joker tidak berubah. Jika salah satu atau kedua joker berada di posisi ujung, proses *triple cut* tetap dilakukan.

5. Lakukan *count cut*. Lihat nilai kartu terbawah, anggap nilai tersebut adalah n. Ambil n kartu pertama dan pindahkan ke posisi kedua dari bawah.

Susunan kartu sebelum *count cut* dilakukan:
5 8 11 14 17 20 23 26 **28** 9 12 15 18 21 24 2 **27** 1 4 7 10 13 16 19 22 25 3 6

Susunan kartu setelah *count cut* dilakukan:
23 26 **28** 9 12 15 18 21 24 2 **27** 1 4 7 10 13 16 19 22 25 3 5 8 11 14 17 20 6

6. Temukan kartu keluaran. Untuk melakukan ini, lihat kartu paling atas. Hitung sebanyak nilai kartu tersebut mulai dari kartu setelah kartu paling atas. Nilai kartu pada urutan tersebut adalah nilai berikutnya dalam kunci aliran.

Pada contoh yang digunakan, nilai kartu paling atas adalah 23. Nilai kartu ke 23 dari kartu setelah kartu paling atas adalah 11. Nilai 11 inilah yang dimasukkan ke dalam kunci aliran.

Setelah itu ulangi lagi dari langkah kedua sampai keenam. Lakukan terus sampai sebanyak panjang kunci yang digunakan. Sebelum mengulang langkah-langkah tersebut, urutan kartu dari proses sebelumnya tidak perlu diubah.

2.2. Enkripsi dan Dekripsi

Untuk melakukan enkripsi, yang pertama dilakukan adalah membuang semua karakter non huruf dan ubah semua huruf menjadi huruf kapital. Supaya konsisten dengan praktik kriptografi tradisional, tambahkan huruf 'X' jika dibutuhkan supaya total jumlah karakter menjadi kelipatan 5. Ubah semua huruf tersebut menjadi angka (A=1, B=2, dst).

Gunakan algoritma Solitaire Cipher untuk mendapatkan kunci aliran. Jumlahkan masing-masing pasangan plainteks dan kunci aliran, kemudian lakukan modulo 26 pada hasilnya. Angka yang dihasilkan kemudian diubah kembali menjadi huruf. Huruf-huruf hasil operasi ini lah yang merupakan cipherteks.

Proses dekripsi hanya merupakan kebalikan dari proses enkripsi. Dimulai dengan mengubah cipherteks yang akan didekripsi menjadi angka-angka. Dengan menggunakan tumpukan kartu yang sama persis dengan kartu yang digunakan saat enkripsi, hasilkan kunci aliran. Kurangi nilai-nilai pada cipherteks dengan kunci aliran, kemudian lakukan modulo 26. Untuk memermudah penghitungan jika nilai pada cipherteks kurang dari nilai pada kunci aliran, tambahkan terlebih dahulu 26 kepada nilai di cipherteks. Setelah itu, kembalikan angka-angka hasil operasi menjadi huruf.

Misal plainteks yang akan dienkripsi adalah
ARKAVIDIA
dan kunci aliran yang akan digunakan adalah
4 23 10 24 8 25 18 6 4 9

Plainteks diatur terlebih dahulu sehingga menjadi
ARKAV IDIAX

Berikutnya plainteks tersebut diubah menjadi angka, sehingga mejadi
1 18 11 1 22 9 4 9 1 24

Jumlahkan plainteks yang sudah diubah menjadi angka dengan kunci aliran, kemudian lakukan modulo 26.

```
1 18 11  1 22  9  4 9 1 24
4 23 10 24  8 25 18 6 4 9  +
5 15 21 25  4  8 22 15 5 7
```

Konversikan kembali angka-angka tersebut menjadi huruf.
E O U Y D H V O E G

Cara untuk mengembalikan cipherteks tersebut kembali menjadi plainteks pertama-tama dengan mengkonversi cipherteks menjadi angka.
5 15 21 25 4 8 22 15 5 7

Temukan kunci aliran yang digunakan. Untuk menemukan kunci aliran yang benar, susunan awal kartu harus benar-benar sama persis. Dalam contoh ini yang digunakan adalah
4 23 10 24 8 25 18 6 4 9

Kurangi cipherteks yang sudah diubah menjadi angka dengan kunci aliran, kemudian lakukan modulo 26.

```
5 15 21 25 4   8 22 15 5 7
4 23 10 24 8 25 18   6 4 9  -
1 18 11  1 22  9   4   9 1 24
```

Konversi kembali angka-angka tersebut menjadi huruf, sehingga menjadi

ARKAVIDIAX

Buang huruf 'X' jika tidak diperlukan. Pada contoh ini menjadi

ARKAVIDIA

3. Kelebihan dan Kekurangan Solitaire Cipher

Kelebihan:

1. Solitaire Cipher ini memiliki tingkat keamanan yang bisa diandalkan. Bahkan bisa disejajarkan dengan One Time Pads jika algoritma ini menggunakan kunci sepanjang plainteks. Hal ini disebabkan pembangkitan *keystream* dari urutan kartu melalui proses yang panjang sehingga hasil keluarannya benar-benar acak.

2. Algoritma Solitaire tidak membutuhkan alat komputasi tertentu. Algoritma ini hanya membutuhkan alat tulis dan satu pak kartu.

3. Proses yang dilakukan untuk mengenkripsi sebuah karakter cukup rumit sehingga tidak mudah untuk dianalisis oleh seorang kriptanalis.

Kekurangan:

1. Proses enkripsi dan dekripsi memakan waktu yang sangat lama. Untuk mengenkripsi pesan yang lumayan panjang waktu yang dibutuhkan bisa mencapai satu hari.

2. Kesalahan pada saat proses enkripsi dan dekripsi sangat mungkin terjadi. Jika terjadi kesalahan pada satu langkah maka akan mengacaukan pesan selanjutnya.

3. Kemungkinan berulangnya nilai keluaran untuk setiap tahap pembangkitan *keystream* cukup tinggi.

4. Jika panjang keystream yang digunakan pendek, maka algoritma ini bisa dipecahkan dengan metode Kasiski.

4. Perbandingan dengan Algoritma Enkripsi lain

4.1. Vigenere Cipher

Vigenere Cipher menggunakan Bujursangkar Vigenere untuk melakukan enkripsi. Kolom paling kiri dari bujursangkar menyatakan huruf-huruf kunci, sedangkan baris paling atas menyatakan huruf-huruf plainteks.

Setiap baris di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh dengan Caesar Cipher, yang mana jauh pergeseran huruf plainteks ditentukan nilai desimal oleh huruf kunci tersebut.

Vigenere Cipher dipublikasikan pada tahun 1856, tetapi algoritma tersebut baru dikenal luas 200 tahun kemudian dan berhasil dipecahkan oleh Babbage dan Kasiski pada pertengahan abad 19.

Pada bagian ini, algoritma Solitaire Cipher ini akan dibandingkan dengan algoritma Vigenere, yang sama-sama merupakan algoritma berbasis substitusi. Ada tiga bagian yang akan dibahas, yaitu tingkat keamanan, proses enkripsi dan proses dekripsi, serta penggunaan kunci.

Untuk perbandingan tingkat keamanan, algoritma Solitaire Cipher lebih unggul karena algoritma ini menggunakan keystream yang panjangnya sama dengan plainteks. Kunci yang lebih panjang menjamin algoritma ini lebih susah untuk dipecahkan. Sedangkan algoritma Vigenere yang menggunakan kunci yang lebih pendek bisa dipecahkan dengan menggunakan teknik analisis frekuensi dan metode Kasiski.

Perbandingan kedua adalah dalam proses enkripsi dan dekripsi. Pada algoritma Solitaire, proses enkripsi dan dekripsi ini terdiri dari enam tahap yang lumayan rumit. Sebaliknya pada algoritma Vigenere proses enkripsi dan dekripsi hanya terdiri dari satu langkah penggantian karakter dari tabel substitusi yang ada. Proses dekripsi pada kedua algoritma hampir sama dengan proses enkripsi yang dilakukan.

Hal ketiga yang dijadikan perbandingan adalah penggunaan kunci dalam proses enkripsi. Algoritma Vigenere menggunakan kunci untuk mencari posisi karakter pengganti pada tabel. Sedangkan pada algoritma Solitaire, kunci digunakan untuk menginisiasi urutan kartu yang

kemudian diacak lagi dengan proses pembangkitan kunci aliran(*keystream*).
Pada dasarnya, yang membedakan algoritma Solitaire cipher dengan algoritma Vigenere Cipher adalah proses pencarian kunci aliran. Jika pada Vigenere Cipher kunci aliran adalah kunci yang disepakati kedua belah pihak, pada Solitaire Cipher kunci aliran harus dicari terlebih dahulu berdasarkan susunan kartu yang disepakati oleh kedua belah pihak. Selain itu, Solitaire Cipher benar-benar sama dengan Vigenere Cipher standar.

4.2. One-Time Pad

One-Time Pad ditemukan pada tahun 1917 oleh Major Joseph Maughborgne. Cipher ini termasuk ke dalam kelompok algoritma kriptografi simetri.

One-time pad (pad=kertas bloknot) berisi deretan karakter-karakter kunci yang dibangkitkan secar acak. Aslinya, satu buah one-time pad adalah sebuah pita (tape) yang berisi barisan karakter-karakter kunci.

Satu pad hanya digunakan sekali (one-time) saja untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain.

Aturan enkripsi yang digunakan persis sama seperti pada Vigenere Cipher. Pengirim pesan menggunakan setiap karakter kunci untuk mengenkripsikan satu karakter plainteks. Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci one-time pad.

Panjang kunci yang digunakan sama dengan panjang plainteks, sehingga tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi. Setelah pengirim mengenkripsikan pesan dengan kunci, ia menghancurkan kunci tersebut.

Untuk perbandingan tingkat keamanan, algoritma Solitaire Cipher lebih unggul karena algoritma ini menggunakan kunci aliran yang tidak bisa dibaca langsung oleh pihak lawan. Kerahasiaan algoritma penghasil kunci aliran membuat kunci aliran tidak bisa diketahui oleh sembarang orang. Posisi awal deck yang sangat penting untuk menghasilkan kunci aliran juga dirahasiakan. Sedangkan pada one-time pad, jika jatuh ke tangan pihak lawan, maka pihak lawan dapat dengan mudah mendekripsi cipherteks.
Perbandingan kedua adalah dalam proses enkripsi dan dekripsi. Pada algoritma Solitaire, proses enkripsi dan dekripsi ini terdiri dari enam tahap yang lumayan rumit. Sebaliknya pada algoritma one-time pad proses enkripsi dan dekripsi hanya terdiri dari satu langkah penggantian karakter yang dapat juga dilakukan dengan menggunakan tabel substitusi yang ada, seperti pada Vigenere Cipher. Proses dekripsi pada kedua algoritma hampir sama dengan proses enkripsi yang dilakukan.

Hal ketiga yang dijadikan perbandingan adalah penggunaan kunci dalam proses enkripsi. One-time pad menggunakan kunci untuk mencari karakter pengganti. Sedangkan pada algoritma Solitaire, kunci digunakan untuk menginisiasi urutan kartu yang kemudian diacak lagi dengan proses pembangkitan kunci aliran(*keystream*).

5. Kesimpulan

Dari hasil eksplorasi mengenai Solitaire Cipher ini, ada beberapa kesimpulan yang dapat diambil.

1. Solitaire sebagai salah satu algoritma kriptografi sederhana mempunyai tingkat keamanan yang cukup tinggi jika dibandingkan dengan algoritma sederhana lain. Hal ini karena pada algoritma Solitaire ini sebuah karakter tidak selalu dienkripsikan menjadi satu karakter tertentu seperti algoritma kriptografi klasik lain.

2. Solitaire Cipher ini cocok digunakan untuk penyampaian pesan karena tidak membutuhkan alat komputasi tertentu, hanya membutuhkan satu pak kartu remi dan seperangkat alat tulis.

3. Proses enkripsi dan dekripsi Solitaire Cipher membutuhkan waktu yang sangat lama. Proses pembangkitan *keystream* yang berulang adalah proses yang paling banyak memakan waktu.

4. Algoritma Solitaire Cipher bisa dipecahkan dengan metode Kasiski jika panjang keystream yang digunakan tidak sama dengan panjang plainteks. Jika sama dengan panjang plainteks, algoritma ini justru hampir sebanding dengan One-time Pads sebagai *cipher* yang tidak bisa dipecahkan.

5. Algoritma Solitaire ini tidak sepenuhnya menghasilkan keystream yang acak. Peluang

sebuah huruf untuk muncul di proses pembangkitan aliran kunci selanjutnya cukup besar.

Algoritma ini masih memiliki kekurangan dalam hal pengacakan huruf keluaran dan juga waktu yang digunakan. Tetapi algoritma ini sangat potensial untuk digunakan karena selain tidak membutuhkan alat tertentu juga tingkat keamanannya yang lumayan tinggi. Perbaikan dalam proses pembangkitan aliran kunci baik itu mengurangi atau menukar urutan proses akan meminimalisir kemungkinan munculnya huruf yang sama secara berurutan. Perbaikan yang tidak kalah penting adalah optimalisasi waktu yang digunakan untuk proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

[1]http://www.schneier.com/solitaire.html

[2]http://bradconte.com/security/solitaire_cipher.php

[3]http://rubyquiz.com/quiz1.html

[4]http://bradconte.com/security/solitaire_cipher.php

[5]http://www.b-con.us/security/

[6]http://www.ciphergoth.org/crypto/solitaire/

[7]http://www.cryptography.mesogunus.com/

[8]http://home.earthlink.net/~neilbawd/solitair.html

[9]http://www.jera.com/solitaire/

[10]http://www.ussrback.com/crypto/misc/solitaire.html

[11]http://www.wikipedia.org/Solitaire_(cipher)