Pengukuran dan Pengujian Kekuatan Algoritma Auto-key Vegenere Cipher

Timotius Triputra Safei (13509017)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509017@stei.itb.ac.id

***Abstract*—Kriptografi merupakan seni mengamankan data yang pada saat ini merupakan salah satu isu penting dalam dunia informatika. Cara yang digunakan dalam kriptografi cukup beragam, dimulai dari kriptografi klasik samapai kriptografi modern. Salah satu teknik dalam kriptografi klasik adalah vigenera cipher. Algoritma ini mempunyai kemiripan dengan Caesar cipher. Algoritma ini mengenkripsi data dengan cara menggeser huruf sesuai dengan key. Salah satu bentuk pengembangan dari vigenere cipher ini adalah auto-key vigenere. Algoritma ini menggunakan plaintext yang didapat sebagai key.**

Makalah ini berisikan hasil studi dan percobaan mengenai auto-key vigenere cipher. Sampai saat ini, sudah ditemukan berbagai cara pemecahan vigenere cipher, akan tetapi diperkirakan untuk pemecahan auto-key vigenere masih merupakan hal yang cukup sulit. Oleh karena itu, dalam makalah ini akan dibahas mengenai tingkat kekuatan dari auto-key vigenere cipher beserta pengujiannya. Diaharapkan dari makalah ini dapat ditarik kesimpulan yang valid mengenai tingkat kekuatan auto-key vigenere cipher.

***Index Terms*—Caesar cipher, key, vigenere cipher, auto-key vigenere cipher.**

I. INTRODUCTION

Kriptografi adalah ilmu dan seni untuk menjaga keamanan dari data maupun pesan yang kitamiliki. Kriptografi pada masa lalu lebih ditekankan pada menjaga kerahasiaan dari pesan yang dikirim pengirim kepada penerima. Cara yang dimaksud adalah mengganti pesan yang da ke dalam suatu sandi sehingga tidak dapat dimengerti maksudnya. Akan tetapi, pada masa kini, pesan yang dimaksud dapat berbentuk apapun, dalam berbagai media. Oleh karena itu, cara-cara baru terus dicoba dan digunakan.

Di dalam kriptografi ada beberapa unsur keamanan penting yang menjadi pelayanan dari kriptografi. Beberapa unsur tersebut adalah sebagai berikut:

1. *Data Confidentiality*
   Salah satu aspek keamanan utama dari kriptografi tentu saja kerahasiaan dari pesan yang kita miliki. Dalam kriptografi, salaha satu tujuan utaqma adalah berusaha menjaga kerahasiaan pesan yang kita miliki. Pesan yang ada tidak boleh diketahui sembarang orang. Oleh karena itu, kerahasiaan pesan juga termasuk menjaga agar hanya orang yang diperbolehkan yang dapat melihat isi dari pesan tersebut.

2. *User Authentication*
   Aspek ini berkaitan dengan siapa pengirim pesan tersebut. Pertanyaan utama dari aspek ini adalaha apakah pesan tersebut adalah pesan yang berasal pihak yang benar-benar mempunyai izin ataukan ada pihak lain yang sengaja mengirim pesan yang palsu.

3. *Message Authentication*
   Dalam pengiriman pesan, sangat mungkin terjadi modifikasi pesan, baik secara tidak sengaja ataupun disengaja. Oleh karena itu, salah satu layanan kriptografi adalah bagaimana menjaga keaslian pesan yang kita kirim ataupun yang kita terima.

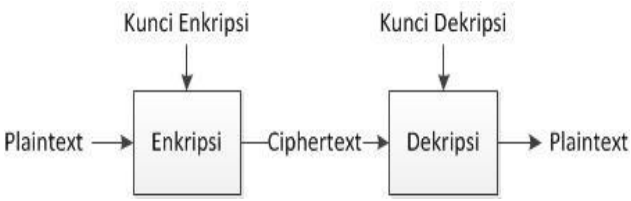
4. *Nonrepudiation*
   Layanan lainnya dari kriptoghrafi adalah anti penyangkalan. Dalam prosesnya kriptografi mengupayakan agar pesan yang dirubah tidak dapat disangkal oleh pengirimnya.

Kriptografi sudah digunakan bahkan sejak 400 SM. Oleh karena itu, kriptografi sudah berkembang sedemikian hingga memiliki berbagai metode serta berbagai perkembangan.

Salah satu perkembangan kriptografi adalah penggunaan key. Pada masa lalu, kriptografi lebih mengutamakan pemakaian algoritma yang dirahasiakan untung menyimpan pesan. Apabila algoritma itu “bocor” atau diketahui pihak lain, maka seluruh pesan dapat dibaca dan algoritma tersebut harus dibuang dan diganti. Hal in tentu saja tidak mangkus karena untuk menjamin kerahasiaan pesan, perlu membuat algoritma terus-menerus dan menjaga agar algoritma tersebut tidak bocor.

Kriptografi pada zaman modern sudah menggunakan key dan kerahasiaan yang perlu dijaga adalah kerahasiaan key yang digunakan Pada masa ini, sudah banyak algoritma yang dibuat dan hasil penyembunyian pesannya tergantung dari kunci atau key yang digunakan. Hal ini tentu jauh lebih mangkus karena pengirim pesan tidak perlu terus membuat algoritma baru. Bahkan algoritma yang dibuat biasanya bersifat *public* dan bebas digunakan serta diketahui oleh semua orang.

Proses utama dalam kriptografi ada dua, yaitu enkripsi dan dekripsi. Enkripsi adalah proses penyembunyian pesan dengan menggunakan key tertentu. Sedangkan dekripsi adalah proses pembacaan atau ekstrasi pesan dari *ciphertext.* Berikut ini gambaran umum dari proses tersebut.



Berikut ini penjelesana mengenai istilah dan component utama yang sering dipakai dalam kriptografi:

1. *Plaintext*
   Plaintext adalah pesan yang akan kita kirim atau simpan dalam bentuk aslinya. Plaintext dapat dibaca secara langsung dan bermakna.

2. *Ciphertext*
   Ciphertext adalah pesan yang sudah kita enkripsi. Ciphertext tidak dapat dibaca secara langsubg dan tidak bermakna.

3. *Enkripsi*
   Enkripsi adalah proses penyembunyian pesan. Proses enkripsi merubah pesan plaintext menjadi ciphertext yang tidak bermakna. Pada algoritma saat ini, untuk melakukan enkripsi diperlukan suatu kunci.

4. *Dekripsi*
   Dekripsi adalah proses mengekstraksi pesan yang ada dalam ciphertext. Proses dekripsi akan menghasilkan plaintext yang sama seperti sebelum dienkripsi. Dalam dekripsi diperlukan juga kunci.

5. *Key / kunci*
   Key adalah suatu parameter yang digunakan untuk melakukan enkripsi maupun dekripsi. Kunci yang digunkan dapat berbentuk apapun seperti abjad, bilangan, atau bahkan dalam kriptografi modern dapat berupa bit.

Dalam perkembangannya, dikenal dua jenis kriptografi berdasarkan penggunaan kunci. Kriptografi simetri, yaitu kriptografi yang kunci untuk enkripsi dan dekripsinya sama dan kriptografi nirsimetri, yaitu algoritma yang menggunakan 2 jenis kunci.

Lewat notasi, proses tersebut dapat ditulis sebagai berikut:

Enkripsi

$$\mathbf{E_k(P) = C}$$

E = fungsi enkripsi
P = plaintext
C = ciphertext
K = key

Dekripsi

$$\mathbf{D_k(C) = P}$$

D = fungsi dekripsi
P = plaintext
C = ciphertext
K = key

Karena plaintext yang dihasilkan dari dekripsi seharusnya sama dengan plaintext awal, maka persamaan tersebut dapat ditulis sebagai berikut:

$$\mathbf{D_k(E_k(P)) = P}$$

E = fungsi enkripsi
D = fungsi dekripsi
P = plaintext
C = ciphertext
K = key

Kekuatan algoritma cipher (algoritma kriptografi klasik) sangat bergantung pada key. Oleh karena itu perlu dilakukan memilihan key dengan baik. Ada dua syarat yang harus dipenuhi dalam memilih key agar terbentuk *unbreakable cipher.* Styarat tersebut adalah :

1. Pemilihan key harus benar-benar acak
2. Panjang key harus sama dengan plaintext.

Apabila kedua syarat tersebut dapat terpenuhi maka ciphertext yang dihasilkan tidak miungkin dipecahkan.

II. VIGENERE CIPHER

Salah satu algoritma kriptografi klasik adalah vigenere cipher.Algoritma ini mirip seperti Caesar cipher, akan tetapi menggunakan key. Inti dari algoritma ini adalah menggeser karakter sebanyak key ke kanan. Dalam notasi matematika dapat ditulis sebagai berikut

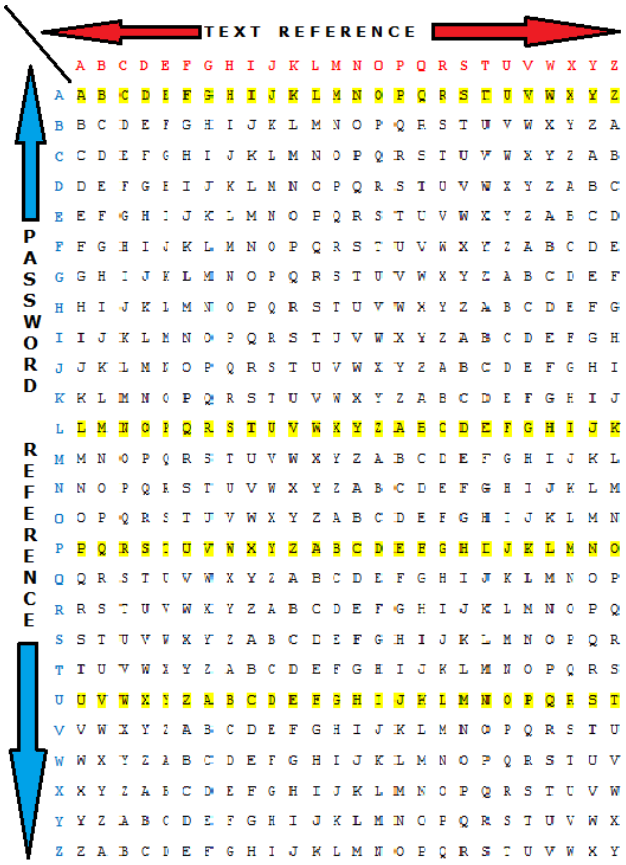
$$\mathbf{C_i = (P_i + K_i) \bmod 26}$$
$$\mathbf{P_i = (C_i - K_i) \bmod 26}$$

C_i = karakter cipher ke i
P_i = karakter plain ke i
K_i = key ke i

Jadi pemetaan karakter tergantung kepada key yang digunakan. Apabila key yang digunakan tidak sepanjang plaintext maka akan dilakukan perulangan. Sebagai contoh, kalimat berikut akan dienkripsi dengan key "kunci".

P = kriptografisangatmenarik
K = kuncikuncikuncikuncikunc
C = ulvrbyaecnsmnpoknzgvklvm

Dari semua kunci yang mungkin digunakan dapat dibuat suatu tabel yang berfungsi sebagai kamus dari tiap karakter. Berikut ini tabel yang menggambarkan semua pemetaan huruf yang mungkin terjadi.



Kelebihan utama dari vigenere cipher ini adalah dapat menyembunyikan frekuensi kemunculan karakter yang menjadi kelemahan dari substitution cipher. Oleh karena itu, vigenere cipher lebih kuat dibandingkan substitution cipher biasa.

Akan tetapi, vigenere cipher tetap memiliki kelemahan. Kelemahan inilah yang digunakan oleh Friedrich Kasiski untuk menemukan cara memecahkan vigenere cipher.

Pada vigenere cipher biasa, kunci yang digunakan akan terus diulang sampai semua plaintext selesai dienkripsi. Oleh karena itu akan terjadi perulangan kunci yang digunakan. Karena perulangan tersebut sangat mungkin terjadi perulangan antara karakter yang sama dan penggunaan kunci yang sama seperti pada contoh berikut.

Plainteks : crypto is short for cryptography
Kunci : abcdab cd abcda bcd abcdabcdabcd
Cipherteks : **csastp** kv siqut gqu **csastp**iuaqjb.

Dari contoh tersebut dapat dilihat kemungkinan terjadinya perulanagn. Selain itu, vigenere cipher juga lemah terhadap perulangan kata yang sangat sering dipakai seperti dalam bahasa Inggris kata the, th, dan lain-lain. Dengan memanfaatkan hal tersebut maka ditemukanlah metode Kasiski yang digunakan untuk menentukan panjang kunci. Dapat dilihat bahawa perulangan terjadi pada karakter dengan urutan kelipatan panjang kunci.

Apabila panjang kunci sudah diketahui, maka cipher text tersebut dapat dipecahkan dengan menggunakan analisis frekuensi.

III. Auto-key Vigenere Cipher

Auto-key vigenere cipher adalah pengembangan dari vigenere cipher. Seperti yang sudah diterangkan sebelumnya, vigenere cipher memiliki beberapa kekurangan. Oleh karena itu dicoba pengembangan baru untuk menghilangkan kelemahan tersebut. Kelemahan pada vigenerer cipher biasa, terletak pada perulangan kunci yang mengakibatkan perulangan atau munculnya frekuensi kemunculan dari suatu deret karakter. Untuk menghilangkan hal tersebut maka digunakanlah auto-key.

Auto-key ini, selain menyamarkan frekuensi kemunculan, tetapi juga merupakan usaha untuk mencapai *unbreakable cipher* yang mempunyai 2 syarat yang sudah disebutkan sebelumnya. Dengan menggunakan auto-key, maka didapat kunci yang memiliki panjang sama dengan plaintext.

Cara kerja dari auto-key vigenere cipher adalah penggunaan plaintext itu sendiri untuk menjadi key. Jadi, key yang dimasukkan akan digunakan sebagai karakter awal key. Apabila plaintext yang ada lebih panjang dari key, maka plaintext tersebut akan dimasukkan sebagai key. Oleh karena itu, pada saat sebelum dienkripsi, maka akan dilakukan pembangkitan key sesuai dengan panjang plaintext. Berikut ini contoh penggunaannya.

Plainteks : crypto is short for cryptography
Kunci : abcdcryptoisshortforcryptogr
Cipherteks : csasvfghlvwjlmcivwmgvfegtdnp

IV. Pengujian Auto-key Vigenere

A. PESAN

Untuk mengetahui sejauh mana kekuatan dari auto-key

vigenere, maka kita akan mencoba melakukan serangan terhadap ciphertext yang dibuat dari suatu text yang dienkripsi menggunakan auto-key vigenere cipher.

Text yang digunakan berbahasa Inggris. Serangan akan dilakukan beberapa kali dengan beberapa jenis keadaan. Berikut ini text yang digunakan:

"There are no more famous ancient sites within Egypt, or for that matter elsewhere in the world, than the Great Pyramids at Giza. They are, without question, the icon most associated with the Egypt. They have been both the main destination for tourists, and a source of imaginative thought to the world for over three thousand years. "

Text tersebut akan dienkripsi dengan algoritma auto-key vigenere cipher. Untuk kunci yang digunakan, dipakai kata "CODE". Hasil dari plaintext tersebut adalah ciphertext sebagai berikut:

"VVHVX HVVRO DSESR ODSZS MBWAE AVAMG XKEBX ZEVXN GCXUP UHFKM OKFHT MQRXE WVASW VAPRK LMJHY PZHYL QMOET KLEZG CRTBG USMBJ AZTZP DYTYI UIKLK CMXIY LJCSF MPSVV VRUQG GMGKH CASHG LWBXK PPXLZ FTXZF TRAHZ CIEZR CSXUU VXTTP RPEAG LRSMQ BNYWF GTIIB GNJIF WSSBX RUSIW KQOLQ ZAZQI EMPJY ZOHNU AAXKH YPZTF CRASI HCVVX AYSYL HBXQE NUQ"

B. SERANGAN PERTAMA

Serangan pertama yang akan dicoba adalah menggunakan metode kasiski. Penggunaan ini digunakan untuk membandingkan auto-key vigenere cipher dan vigenere cipher biasa.

Dari ciphertext yang ada, dilakukan analisis berupa perhitungan frekuensi n-graph. Terdapat beberapa data yang menarik sebagai berikut:

VVR	6,129,402
GCX	40, 307, 626
BGU	93,299,480,614
CDWRLBGU	475, 609

Dapat dilihat, bahwa kemunculan n-graph masih sangat mungkin terjadi. Akan tetapi, apabila kita menganalisa lebih lanjut, metode kasiski ini tentu saja tidak dapat digunakan karena untuk langkah lebih lanjut, diperlukan perhitungan frekuensi untuk setiap factor dari panjang kunci kunci.

Dalam kasus auto-key vigenere cipher, tentu saja analisis frekuensi untuk tiap karakter dari kunci tidak mungkin dilakukan karena kunci hanya dipakai sekali. Meskipun masih mungkin terjadi perulangan bahkan pada kasus ini samapa 8-graph yang biasanhya sangat menjanjikan untuk digunkanan dalam metode kasiski. Oleh karena itu, meskipun kita menebak panjang kunci yang digunakan, kelompok-kelompok kata yang terbentuk selanjutnya tidak dapat dinalisis frekuensinya karena bukan merupakan ciphertext dengan 1 karakter key.

C. SERANGAN KEDUA

Serangan kedua adalah serangan yang dilakukan dengan kondisi kita hanya mengetahui cipher text yang ada dan bahasa yang digunakan. Dalam serangan ini, kita mencoba untuk melihat kemungkinan letak kata-kata yang hampir pasti dipakai.

Dalam penggunaan auto-key vigenere cipher, plaintext digunakan sebagai kunci untuk enkripsi, oleh karena itu, pasti terdapat kata-kata yang hampir pasti digunakan dalam bahasa yang digunakan. Dengan menebak kata tersebut, maka bisa dianggap kita mendapatkan sebagian kunci akan tetapi kita belum tahu di mana letak kunci tersebut.

Oleh karena itu, hal selanjutnya yang kita lakukan adalah mencoba secara *brute force* untuk menemukan letak kunci tersebut. Selain itu, kita juga perlu melakukan beberapa asumsi untuk mempermudah pencarian seperti perkiraan *range* panjang kunci yang digunakan.

Karena kita ketahui bahwa plaintext merupakan bahasa Inggris, maka terdapat beberapa kata yang bisa dipilih. Akan tetapi, semakin panjang kata yang dipilih semakin baik karena akan mudah dianalisis dan ditebak. Kata-kata tersebut biasanya merupakan kata hubung ataupun kata-kata umum, contohnya the, this, that, there. Apabila kita mengetahui tema dari plaintext, maka akan lebih mudah untuk mencari kata sepanjang-panjangnya. Sebagai contoh kali ini kita akan menggunakan kata "that". Berikut ini langkah-langkah yang diambil.

- Kita menentukan kata yang akan kita pakai sebagai kunci, pada kasus ini kita memakai kata that.

- Iterasi kunci "that" pada plaintext dari awal sampai akhir. Seperti berikut

  Ciphertext : VVHVX HVVRO DSESR
  Key : that. ….. …..
  Plaintext : cohc. ….. …..

  Ciphertext : VVHVX HVVRO DSESR
  Key : .ther ..... …..
  Plaintext : .cave .…. …..

  Ciphertext : VVHVX HVVRO DSESR
  Key : ..tha t... …..
  Plaintext : ..oox o.… …..

  Ciphertext : VVHVX HVVRO DSESR
  Key : …th at.. …..
  Plaintext : …cq hc... …..

dst…

Ciphertext : UHFKM OKFHT MQRXE
Key : ….. .that …..
Plaintext : ….. .ryha …..

Ciphertext : UHFKM OKFHT MQRXE
Key : ….. ..tha t....
Plaintext : ….. ..mat t....

dst..

- Dari hail iterasi tersebut, urutkan hasil yang didapat dari yang paling mungkin sampai yang paling tidak mungkin.

  Cqhc, cohc,…,…,……..,matt,cave,….

- Pilih salah satu hasil dan anggap bahwa hasil itu benar. Pada kasus ini kita ambil kata matt.

- Setelah itu, coba cari plaintext dari sekitar tempat key yang ditentukan dan geser huruf demi huruf untuk melihat panjang key.

  Panjang kunci 4
  Ciphertext : UHFKM OKFHT MQRXE
  Key : ….. ..tha tmatt
  Plaintext : …th atmat terel

  Panjang kunci 5
  Ciphertext : UHFKM OKFHT MQRXE
  Key : ..mdm v.tha t.mat
  Plaintext : ..tha t.mat t.fxl

  Panjang kunci 6
  Ciphertext : UHFKM OKFHT MQRXE
  Key : .oykt ..tha t..ma
  Plaintext : .that ..mat t..le

- Dari hasil tersebut bias kita lihat panjang kunci yang paling mungkin adalah 4. Oleh karena itu kita teruskan dengan asumsi panjang kunci 4.

  Ciphertext:
  UHFKMOKFHTMQRXEWVASWVAPRK
  Key : ptorf ortha tmatt erels ewher
  Plaintext : forth atmat terel sewhe reint
  dst…

- Pada akhirnya seluruh plaintext dapat diekstrak dan kunci yang ditemukan adalah “code”

D. SERANGAN KETIGA

Serangan yang ketiga dilakukan berdasarkan analisis kelemahan dari auto-key vigenere cipher. Diperkirakan bahwa kelemah utama dari auto-key vigenere cipher adalah known plaintext attact.

Karena menggunakan plaintext sebagai kunci, maka apabila ada bagian plaintext yang diketahui, maka sama saja dengan kita mengetahui sebagian dari kunci. Kriptanalisis bahkan akan semakin mudah jika kata yang diketahui relatif panjang dibandingkan kunci dan jarang digunakan dalam bahsa sehari-hari.

Pada kasus kali ini kita sumsikan bahka kita mengetahui bahwa terdapat kata “imaginative” pada plaintext. Maka kita dapat melakuikan serangan seperti berikut.

- Lakukan enkripsi pada kunci secara iteratif

  Panjang key : 1
  Plaintext : imaginative
  Key : .imaginativ
  Ciphertext : UMGOV NTBDZ

  Panjang key : 2
  Plaintext : imaginative
  Key : ..imaginati
  Ciphertext : ISITI GIOM

  Panjang key : 3
  Plaintext : imaginative
  Key : …imaginat
  Ciphertext : OUNGB VVX

  Panjang key : 4
  Plaintext : imaginative
  Key : ….imagina
  Ciphertext : QZAZQ IE

  Panjang key : 5
  Plaintext : imaginative
  Key : …..imagin
  Ciphertext : VMTOD R

- Cari potongan ciphertext tersebut pada ciphertext. Semakin panjang kesamaan yang ditemukan, semakin besar kemungkinan kita mendapatkan kata yamh dimaksud. Oleh karena itu, pencarian dimulai dari panjang key terkecil dan akhirnya didapatkan pada panjang key 4.

  KQOL**Q ZAZQI E**MPJY ZOHNU

- Selanjutnya lakukan dekripsi berdasarkan kunci dan letaknya.

  Ciphertext : KQOL**Q ZAZQI E**MPJY ZOHNU
  Key : . . . . i magin a…. …..
  Plaintext : …. i nativ e…. ….

  Ciphertext : KQOL**Q ZAZQI E**MPJY ZOHNU
  Key : . . . . i magin ative …..
  Plaintext : imagi nativ e…. ….

Ciphertext : KQOL**Q ZAZQI E**MPJY ZOHNU.
Key : ceofi magin ative thoug
Plaintext : imagi nativ ethou ghtto

dst…

- Dari potongan tersebut, kita dapat mendapatkan keseluruhan potongan plaintext.

V. ANALISIS

Dari beberapa serangan yang sudah dilakukan, kita bisa menganalisa beberapa kekurangan dan kelebihan dari auto-key vigenere cipher.

Sebagaimana tujua awal dari pengembangan vigenere cipher, auto-key vigenere cipher dapat menghilangkan kelemahan vigenere cipher biasa. Auto-key vigenere cipher dapat menyembunyikan frekuensi kemunculan potongan cipher yang berulang karena penggunaan kunci secara berulang. Akibat dari hal tersebut, metode kasiski tidak dapat digunakan pada auto-key-vigenere cipher.

Penggunaan plaintext sebagai kunci memang meningkatkan kekuatan algoritma. Hal ini sesuai dengan persyaratan dari *unbreakable cipher*. Akan tetapi, karena tidak semua syarat dipenuhi, maka terdapat kelemahan yang cukup fatal. Kelemahan tersebut adalah mudah ditebaknya key yang digunakan. Dari kelemahan tersebut maka algoritma ini dapat dipecahkan.

Meskipun dapat dipecahkan, tetapi secara keseluruhan algoritma ini masih lebih baik daripada vigenere cipher biasa. Hal ini dapat dilihat dari kompleksitas pemecahan dan waktu yang dibutuhkan. Akan tetapi apabila diserang lewat known plaintext attact yang menjadi kekurangan utama, maka untuk memecahkan algoritma ini tidak terlau sulit.

VI. CONCLUSION

Jika kita melihat dari percobaan yang sudah dilakukan serta analisis maka kita dapat mendapatkan beberapa kseimpulan sebagai berikut:

1. Algoritma auto-key vigenere cipher lebih baik daripada algoritm,a vigenere cipher.
2. Algoritma ini kurang kuat karena meskipun untuk memecahkannya memerlukan *resource* cukup banyak, tetapi jika kelemahan utamanya ditemukan maka sangat mudah untuk memecahkannya.

REFERENCES

[1] http://www.dotnetspider.com/attachments/Resources/41491-3313-Vigenere-cipher-EXAMPLE-TABLE-1.png
[2] http://en.wikipedia.org/wiki/Autokey_cipher
[3] http://www.cryptool-online.org/index.php?option=com_content&view=article&id=105&Itemid=128&lang=en
[4] Munir, Rinaldi, "Diktat Kuliah IF5054 KRIPTOGRAFI", Bandung : Departemen Teknik Informatika ITB, 2005.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Maretl 2012

Timotius T. Safei (13509017)