Modifikasi Vigenere Chipher dengan Penyelipan Huruf Secara Pseudo-Random pada Plainteks

Setia Negara B. Tjaru
13508054
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
Setia.negara.91@gmail.com

***Abstrak*—Vigenere cipher adalah algoritma enkripsi yang jauh lebih baik dari algoritma subtitusi satu-satu seperti Caesar cipher misalnya. Vigenere cipher ini menerapkan Caesar cipher namun pergeseran satu karakter pada tiap barisnya. Hal ini membuat frekuensi kemunculan huruf akan cenderung datar sehingga tidak bisa dipecahkan oleh analisis frekuensi. Namun vigenere cipher ternyata juga memiliki kelemahan, karena kunci yang cenderung lebih pendek sehingga dapat terjadi perulangan plainteks+kunci. Ini dimanfaatkan oleh kasiski untuk menerka panjang kunci. Hal ini biasa ditangani dengan menggunakan kunci yang berasal dari manuskrip atau literatur yang panjang. Pada makalah ini penulis menawarkan cara baru untuk mengelabui teknik kasiski.**

***Index Terms*—Vigenere cipher, Kasiski, plainteks, cipherteks**

I. PENDAHULUAN

Kriptografi adalah suatu cara atau seni untuk menjaga dan menyembunyikan pesan dari pihak yang tidak semestinya. Ada banyak cara untuk melakukan hal ini. Sudah banyak algoritma kriptografi yang ditemukan oleh kriptografer, di antaranya adalah Caesar Cipher dan Vigenere Cipher.

Caesar Cipher adalah cipher yang menyubtitusi satu huruf tertentu dengan huruf lainnya subtitusi dilakukan dengan pergeseran alfabet. Caesar cipher ini sederhana namun dapat dipecahkan dengan brute force ataupun analisisfrekuensi. Salah satu pemakaian cipher ini yang cukup terkenal adalah ROT13. ROT13 adalahCaesar Cipher dengan pergeseran huruf 13 kali. Karena alfabet berjumlah 26, ROT13 dua kali berarti mengembalikannya ke huruf aslinya. ROT13 banyak dipakai pada sistem operasi linux.

Karena Caesar Cipher telah dipecahkan dan memiliki kelemahan, maka para kriptografer membuat algoritma-algoritma baru. Salah satunya adalah Vigenere Cipher. Vigenere Cipher menerapkan Caesar Cipher namun berbeda bergantung pada kuncinya. Vigenere Cipher meminimalkan kemungkinan dipecahkannya cipher dengan analisis frekuensi. Namun telah ditemukan cara memecahkan Vigenere Cipher yaitu dengan analisis kasiski.

II. DASAR TEORI

A. Vigenere Cipher

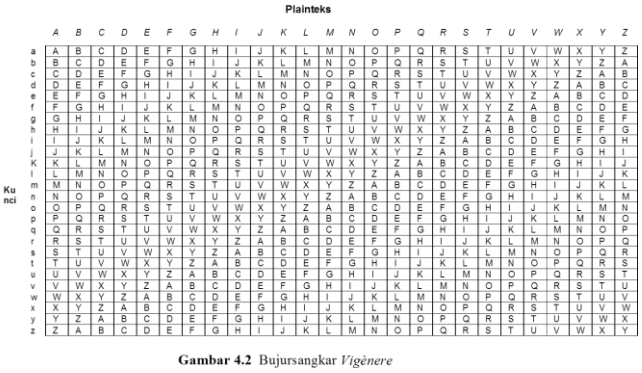
Salah satu permasalahan utama dari Cipher Subtitusi sederhana seperti Caesar Cipher adalah rentannya cipher tersebut terhadap analisis frekuensi. Jika terdapat cipherteks yang cukup maka dengan memeriksa frekuensi huruf yang paling banyak muncul akan memberikan gambaran huruf apa yang merepresentasikan huruf apa pada plainteks.

Oleh karena itu, untuk membuat cipher lebih aman dibuatlah algoritma yang kebal terhadap analisis frekuensi. Di sinilah subtitusi polialfabetik cipher berperan. Dibandingkan subtitusi satu ke satu huruf seperti biasa, terdapat hubungan satu ke banyak antara huruf pada cipherteks dengan huruf palinteks.

Salah satu yang menerapkan subtitusi polialfabetik adalah vigenere cipher. VigenereCipher ini telah dibuat beberapa kali. Pertama kali di temukan oleh Giovan Battista Bellaso dan di jelaskan pada bukunya: La cifra del. Sig. Giovan Battista Bellaso. Namun lebih sering diasosiasikan kepada Blaise de Vigenère. Bahkan namanya pun sekarang disebut Vigenere Cipher.

Cipher ini terkenal karena mudah dimengerti dan diimplementasikan. Karena ini hanya Caesar Cipher yang memakai variabel kunci sebagai indikator pergesarannya. Cipher ini juga terkenal karena terlihat tidak dapat dipecahkan oleh pemula.

Metode enkripsi yang digunakan pada Vigenère Cipher adalah dengan menggunakan tabula recta/Tabel Vigenère, yaitu tabel pergeseran huruf-huruf pada plaintext terhadap huruf-huruf pada kunci.



Gambar 4.2 Bujursangkar *Vigènere*

Gambar 1. Tabula Recta Vigenere

Sebagai contoh, untuk plaintext SETIA NEGARA B TJARU dengan kata kunci TEST, dengan mengacu pada tabel akan menjadi ciphertext sebagai berikut:

plaintext : SETIA NEGARA B TJARU
kunci : TESTT TESTTE S TTEST
ciphertext : LILBT RWZTVS U MNSKN

Vigenère Cipher dapat juga menggunakan rumus sebagai berikut:

$$C_i \equiv (P_i + K_i) \mod 26$$

dimana Ci aDalah huruf ke-i pada ciphertext, Pi adalah huruf ke-i pada plaintext, dan Ki adalah huruf ke-i pada kunci. Setiap huruf tersebut diwakilkan dengan bilangan 0-25 yang masing-masing merupakan enumerasi dari abjad A-Z.

Proses dekripsi pada Vigenère Cipher juga dapat ditemukan dengan menggunakan rumus yang merupakan kebalikan dari rumus enkripsi, yaitu:

$$P_i \equiv (C_i - K_i) \mod 26$$

Jika cipher ini kita terapkan pada karakter ASCII, maka bilangan pembagi adalah 256 sesuai banyak bilangan ASCII. Representasi huruf juga diubah menjadi 0-255.

B. Metode Kasiski

Pada tahun 1863, Kasiski adalah orang pertama yang mempublikasikan kesuksesan penyerangan terhadap Vigenere Cipher. Penyerangan sebelumnya bergantung pada known plaintext atau menggunakan kata yang dapat diketahui sebagai kunci.

Metode Kasiski tidak bergantung pada hal tersebut. Kasiski adalah orang pertama yang mempublikasikan cara dan penyerangan tersebut, namun sebenarnya ada orang lain yang sebelumnya telah menemukannya.

Pada tahun 1854, Charles Babbage telah menemukan cara untuk memecahkan Vigenere Cipher, namun tidak pernah mempublikasikannya. Namun dari catatan studinya terlihat bahwa Babbage menggunakan cara yang nantinya dipublikasikan oleh Vigenere. Bahkan metode tersebut sudah dia pakai sejak 1846.

Metode Kasiski memanfaatkan keuntungan bahwa bahasa Inggris tidak hanya mengandung perulangan huruf, tetapi juga perulangan pasangan huruf atau tripel huruf, seperti TH, THE, dsb. Perulangan kelompok huruf ini ada kemungkinan menghasilkan kriptogram yang berulang.

Contoh:

Plainteks : CRYPTO IS SHORT FOR CRYPTOGRAPHY
Kunci : abcdab cd abcda bcd abcdabcdabcd
Cipher : **CSASTP** KV SIQUT GQU **CSASTP**IUAQJB

Pada contoh ini, CRYPTO dienkripsi menjadi kriptogram yang sama, yaitu CSATP.

Dari hal ini dapat kita simpulkan bahwa secara intuitif: jika jarak antara dua buah string yang berulang di dalam plainteks merupakan kelipatan dari panjang kunci, maka string yang sama tersebut akan muncul menjadi kriptogram yang sama pula di dalam cipherteks.

Hal ini lah yang dimanfaatkan oleh metode kasiski, yaitu mencari dua cipherteks berulang untuk menentukan panjang kunci yang dipakai.

Pada contoh diatas, jarak antara perulangan CSASTP adalah 16. Dengan asumsi bahwa perulangan tersebut berarti plainteks yang berulang maka dapat disimpulkan bahwa kunci yang digunakan panjangnya adalah16, 8, 4, 2, 1.

Biasanya angka 1 dan 2 tidak digunakan karena terlalu pendek. Berarti kita hanya perlu mencoba untuk key dengan panjang 16, 8, atau 4. Semakin panjang teks maka semakin akurat metode kasiski ini karena akan semakin banyakvariasi perulangan.

Ciphertext berikut memiliki dua segmen yang berulang:

Ciphertext:
VHVSSP**QUCE**MRVBVBBB**VHVS**URQGIBDUGRNI CJ**QUCE**RVUAXSSR

Jarak antara VHVS adaah 18. Berarti panjang kunci antara 18, 9, 6, 3, 2, atau 1. Sementara untuk perulangan QUCE jaraknya adalah 30 karakter, berarti panjang kunci adalah 30, 15, 10, 6, 5, 3, 2, atau 1. Dengan mengambil irisan dari himpunan ini dapat kita simpulkan bahwa panjang kunci adalah 6 karena 3, 2, dan 1 biasanya diabaikan karena terlalu pendek.

Setelah panjang kunci diketahui dapat dilakukan banyak cara, salah satunya adalah exhaustive search atau brute force. Jika panjang kunci adalah p, maka jumlah kemungkinan kunci yaitu $26^{p.}$

Kita juga dapat menggunakan Metode Kerckhoff, yaitu membagi ciperteks ke dalam kolom sejumlah panjang kunci. Hal ini berarti tiap kolom dienkripsi dengan satu huruf kunci yang sama. Pada metodenya tiap kolom kemudian di shift sehingga didapatkan kata-kata yang bermakna.

Atau bisa juga dengan menerapkan analisis frekuensi setelah pembagian kolom.

III. PENJELASAN METODE

Karena Metode Kasiski bergantung pada pencarian pengulangan segmen pada cipherteks maka mesti dirancang suatu metode untuk mengatasi kelemahan ini.

Salah satu metode yang ditawarkan adalah menyisipkan huruf secara acak pada plainteks agar plainteks yang berulang akan tersamarkan. Hal ini juga diharapkan mengacaukan jarak perulangan pada plainteks.

Pada metode ini akan digunakan pseudo-random generator untuk menggenerate bilangan acak yang menjadi rentang penyisipan huruf. Rentang penyisipan akan diacak antara 1 huruf sampai panjang kunci huruf. Batas maksimal sepanjang panjang kunci dipilih agar dipastikan tiap perulangan kunci terdapat satu huruf sisipan untuk mengacaukan pola perulangan.

Pada metode ini huruf yang diselipkan sebenarnya bisa tidak bergantung pada apapun karena pada proses dekripsi huruf-huruf ini akan langsung dibuang begitu saja setelah di-decipher.

IV. IMPLEMENTASI

Pada implementasi dibuat aplikasi sederhana yang akan melakukan penyisipan secara pseudo-random. Pseudo-random dibangkitkan dengan menggunakan seed. Penghitungan seed dilakukan dengan rumus sebagai berikut:

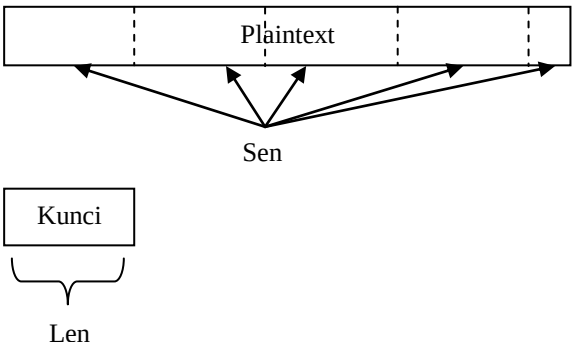
1. Seed = 0
2. Seed ditambah karakter berikutnya
3. Seed dikali karakter berikutnya
4. Seed dikurang karakter berikutnya
5. Ke langkah 2.

```
  public static int hitungSeed(String s){
          int temp = 0;

          for (int i = 0; i < s.length(); i++) {
              if (i % 3 == 0){
                  temp += s.charAt(i);
              } else if (i % 3 == 1){
                  temp *= s.charAt(i);
              } else if (i % 3 == 2){
                  temp -= s.charAt(i);
              }
          }

          return temp;
      }
```

Selanjutnya string plainteks dibagi menjadi beberapa segmen(Sen) yang panjang tiap segmen adalah panjang string kunci. Tiap segmen tadi akan akan dilakukan penyisipan karakter sebanyak 1 karakter sampai panjang_kunci(Len) karakter. Angka 1 dipilih agar tiap segmen yang mungkin berulang minimal ada satu kali penyelipan.



Gambar 2. Skema Penyisipan

```
  public static String selip(String s, String kunci){
          String temp = "";
          int seed = hitungSeed(kunci);
          int len = kunci.length();
          int sen = s.length() / len;

          Random rand = new Random(seed);

          char c[] = s.toCharArray();
          char charArray[][] = new char[sen][len];

          int i = 0;
          int j = 0;
          int k = 0;
          while (i < c.length){
              charArray[j][k] = c[i];
              i++;
              if (i % len == 0){
                  j++;
                  k = 0;
              } else {
                  k++;
              }
          }

          String temp2 = "";
```

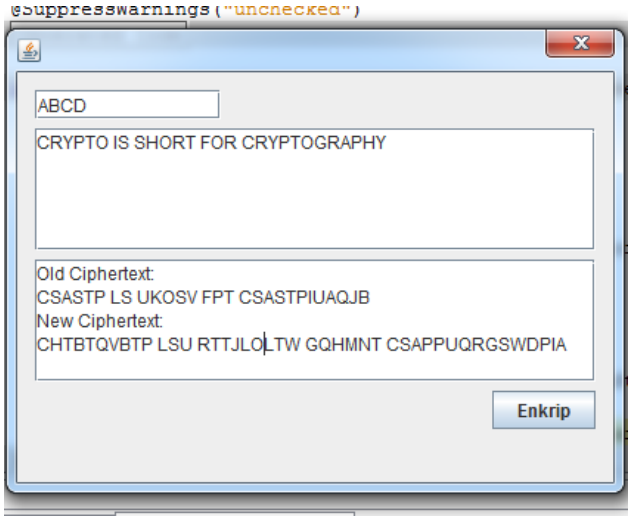
```
            for (int l = 0; l < sen; l++)
{

System.out.println(charArray[l]);
                temp2       =       new
String(charArray[l]);
                int x = rand.nextInt(len-
1) + 1;
                for (int m = 0; m < x;
m++) {
                    int     y           =
rand.nextInt(temp2.length());
                    char    selipan     =
(char)(rand.nextInt(25) + 65);

System.out.println(selipan);
                    temp2       =       new
StringBuffer(temp2).insert(y,
selipan).toString();
                }
                temp += temp2;
            }

            return temp;
    }
```

Dengan demikian hasil implementasi dapat dilihat sebagai berikut:



Karena penyisipan maka ciphertext yang baru menjadi lebih panjang namun sekilas tidak terlihat ada pola perulangan yang sebelumnya tampak sangat jelas yaitu CSASTP.

V. ANALISIS

Pada hasil implementasi terlihat bahwa Ciphertext yang baru tidak lagi terlihat adanya perulangan.

Old Plainteks:
CRYPTO IS SHORT FOR CRYPTOGRAPHY
New Plainteks:
CGRYTPTYTO IST OTSHIOKRT FOEMMR CRYMPTOOGRUAPHY

Old Cipherteks:
CSASTP LS UKOSV FPT CSASTPIUAQJB
New Cipherteks:

Jika Cipherteks lama di analisis dengan kasiski tools:

Kasiski Analysis

ciphertext string	occurs at (index in string)	spacing (number of symbols)	factors
CSASTP	0 16	16	2 4 8 16
SASTP	1 17	16	2 4 8 16
ASTP	2 18	16	2 4 8 16
STP	3 19	16	2 4 8 16

Sedangkan pada Cipherteks baru tidak ditemukan perulangan.

Terlihat di sini bahwa bukan hanya jarak perulangan yang berubah, tapi cipherteks juga berubah jauh sehingga metode kasiski tidak dapat diterapkan lagi.

Untuk teks berukuran besar dengan menggunakan kunci "ABCD"
Plainteks:
ASCII stands for American Standard Code for Information Interchange Computers can only understand numbers so an ASCII code is the numerical representation of a character such as a or or an action of some sort ASCII was developed a long time ago and now the nonprinting characters are rarely used for their original purpose Below is the ASCII character table and this includes descriptions of the first nonprinting characters ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure If someone says they want your CV however in ASCII format all this means is they want plaintext with no formatting such as tabs bold or underscoring the raw format that any computer can understand This is usually so they can easily import the file into their own applications without issues Notepadexe creates ASCII text or in MS Word you can save a file as text only

Old Cipherteks:
ATELI uwaofv gqu Bohrjedn Uwaofdre Foeg fpt Iohrrncwipp Iovhrdjdnhg Cposuugus edn qqlz xnegusucqd pxmcgus ur bp ATELI erdf ls vke pxmftlcbn rfruetgqtbvloo rf c cicuadvhr uxci ds c os rr cq bewipp og vong sptw BUFIJ zat gewgooqgg b oooi tjoh bir bpg oqz ujh oqqpskqtjpj djdrbewesu asg rbthlz xsff fpt tiglr quihkqam susrrsf Eemqz ju tig ATELI ekascftft tbdoe cqd vkit lndnxdfu dfufrjrwippv ph tig fjtvt prnqtlnukqg ekascftftv BUFIJ zat dcuwdlma dfulgogg gqu vuh xkwh vhlfvbpfu aof sp whf geteuiqvloou asg spohwicw pdvcvth Jh spohoog sbav ujhy ydnu bovt CW koxgyes ln CVCJK

fptpau dlm whju mfcqs kv ujhy ydnu slbkqtfzw xkwh pr gqumbvwioi svek bu tbdv cqod qu vpgesufoskqg vke tdw hrrncw ujdt cqy ermqwwes fao xnegusucqd Vkit ls wvubnoy ur ujhy edn gdsjnb jososv tig fjnh jpwo vkejt oxp aqroidcwippv xkwhpww juvufu Npvhpbfhxf frfcwet DSDKL ugat qu jp MT Zosf ypw cbp sbxh b iimg at weyv oonb

New Cipherteks:
ATJHCDKQHJ vHuYdneu XChrWsTTS CpesKlKdcq TDTtJcqKecud YACCKrSeg RYhrIs LSJpLfpXPrncUtjqqQC LnJJweItfhCcqgLg CpDpIXrxTLAwesuB dcq WqVnLRoy wqdQgFrBEvMucqAeA nSwpbfGusM FspNAW cq BUTCVKL Berdf FGju UuNkDf qMvohTsMWidcIlT uVfrMrfuDHfJqtUcwNjSrn qiQ KXa HIcicuaLewesR svMfUi SatMJ b rrI rrLH aSp aBeFDukQXKqq pOXfR vongG EuBosvZ ECVCJKI xcv NfhvJgooqPDBff Ub IKmqqgE wiIohFR dgpK aTPqd WqApDz Lvke RqoUpsrjNOnuRQioIj YekascDcugQrt dOsgG sXdrfnb KwvTPFhd RiLpt VuWkejJu ptROjilRoDdlD OpDwuTqqveZ ZBCCFemqz GkvV DwhWgD XCVCJKZX GfhUcNrbewTftHE vdbYnTeS Aaof tSjlUt lndJZlvfhsS NdfuYcsCPiqvlVWqqsF rf vkeLT fPMlrtTw oXrRoruTjpLtjYSnh GPdZkLVcuadTwLftRsO DSDKPSJ GwbuOM KdDdMwGvcolMGb egvAjiFTogg gYrrH xStg wFkwhBE tfnhKuaseTKv bYGEof spHPO vke LSdMgvcPNuQSkstjqAJouZ btTUf QspoFexjHauS ocufuYth JOi tqRmfZrYog sFcbNtR tiRhySG KxcqYuJ VzqxMs FV HkoKyhXwgu Wkq ECVGDKLUO ioRtpauVZ bDoWm whjLvWH peGcqEt lItJE XvkezH wbpw BrTCmcMiovhNBEat HQwjvk oNJPp GNgGrrWoAMbvwiJpj FuxLMek Dcv DvdJcXvN Peomf Bpt URExneEhrterrLkqWhY tiDhF tdHCyO gqJrnGdt JwBicw VcqRz foOosKvvhrU faoE uoUTdfMuStvdne WFikWsT lsE MuJuGubnQlNa sp whUgb dVGaoF OfNdQKulCma iEUHmqIrru whfZL MhXimgO jpwo HwRIjhisMIU qzCo FaSrSpmkfBUKdtFRloouH GLziujPQpww LkvsvgRMt TNpPweqCddfzhY SfHsgdAugXTCu ATJFBJKJ uIZeyAw pt VQkq TOVD JZPptg zKIov faSp sbxEeL QTb iWjnh buTN XwePIaFu rKoEAlz

Dengan Kasiski Tools didapatkan bahwa pada cipherteks lama:

ATELI	0 96	96	2 3 4 6 8 12 16 24 32 48 96
UWAOF	5 22	17	17
FPT	34 246	212	2 4 53 106 212
WIPP	44 320	276	2 3 4 6 12 23 46 69 92 138 276
WIPPV	320 659	339	3 113 339
IPP	45 161	116	2 4 29 58 116
HYYDNU	452 500	48	2 3 4 6 8 12 16 24 48
VUJHYYDNU	449 497	48	2 3 4 6 8 12 16 24 48
XKWH	386 515	129	3 43 129
TIGFJ	327 633	306	2 3 6 9 17 18 34 51 102 153 306
CQDVKITL	298 596	298	2 149 298
EKASCFTFT	284 346	62	2 31 62
TIG	249 633	384	2 3 4 6 8 12 16 24 32 48 64 96 128 192 384
GSP	169 421	252	2 3 4 6 7 9 12 14 18 21 28 36 42 63 84 126 252
CQD	82 298	216	2 3 4 6 8 9 12 18 24 27 36 54 72 108 216
XNEGUSUCQD	75 589	514	2 257 514
EDN	68 618	550	2 5 10 11 22 25 50 55 110 275 550

Terdapat beberapa angka yang muncul seperti 5, 17, 43, 113, 129, 257, 514. Tetapi hasil yang dominan muncul adalah hasil kelipatan 4. Dimana kita tahu 4 adalah panjang kunci untuk ciphertext ini.

Dengan Kasiski Tools didapatkan bahwa pada cipherteks baru:

ATJ	0 1115	1115	5 223 1115
DNE	14 957	943	23 41 943
WST	22 964	942	2 3 6 157 314 471 942
SCP	26 514	488	2 4 8 61 122 244 488
KDC	33 862	829	829
DCQ	34 115	81	3 9 27 81
JPL	66 550	484	2 4 11 22 44 121 242 484
UTJ	76 548	472	2 4 8 59 118 236 472
TJQ	77 672	595	5 7 17 35 85 119 595
WES	110 235	125	5 25 125
LRO	122 419	297	3 9 11 27 33 99 297
RBE	132 470	338	2 13 26 169 338
FSP	153 649	496	2 4 8 16 31 62 124 248 496
BUT	161 1169	1008	2 3 4 6 7 8 9 12 14 16 18 21 24 28 36 42 48 56 63 72 84 112 126 144 168 252 336 504 1008
DHF	206 903	697	17 41 697
CUA	230 566	336	2 3 4 6 7 8 12 14 16 21 24 28 42 48 56 84 112 168 336
SRS	237 1049	812	2 4 7 14 28 29 58 116 203 406 812
IRR	253 1012	759	3 11 23 33 69 253 759
RRL	254 893	639	3 9 71 213 639
ASP	258 1152	894	2 3 6 149 298 447 894
NGG	280 840	560	2 4 5 7 8 10 14 16 20 28 35 40 56 70 80 112 140 280 560
ECV	290 760	470	2 5 10 47 94 235 470
CVCJK	291 457	166	2 83 166
XCV	297 456	159	3 53 159
PDZ	342 560	218	2 109 218
ZLV	344 501	157	157
VKE	346 529	183	3 61 183
VKE	346 655	309	3 103 309
VKE	346 803	457	457
PTV	403 1130	727	727
UWK	406 756	350	2 5 7 10 14 25 35 50 70 175 350
KVV	447 936	489	3 163 489
AOF	488 992	504	2 3 4 6 7 8 9 12 14 18 21 24 28 36 42 56 63 72 84 126 168 252 504
DJZ	499 1139	640	2 4 5 8 10 16 20 32 40 64 80 128 160 320 640
FHS	504 1103	599	599

Pada cipherteks yang baru setelah dianalisis dengan metode kasiski angka yang muncul snagat bervariasi. Bahkan sampai ada bilangan prima yang sangat besar seperti 457, 727, 599, dan 829. Selain itu terdapat banyak angka seperti 3, 61, 103, 17, 41, 697. Sedangkan angka kelipatan 4 tidak terlihat mendominasi.

VI. KESIMPULAN

1. Metode Kasiski dapat dikelabui dengan cara

melakukan penyisipan huruf-huruf secara acak kepada plainteks. Hal ini selain mengubah jarak plainteks yang berulang tetapi juga mengubah cipherteksnya sehingga metode Kasiski sangat sulit untuk diterapkan.

2. Metode Kasiski sangat bergantung pada pola perulangan dimana plainteks yang sama di enkripsi dengan kunci yang sama. Pada makalah ini diperlihatkan ketergantungan ini membuat Kasiski tidak dapat diterapkan pada plainteks yang sudah tidak memiliki pola berulang.
3. Metode penyelipan menyebabkan panjang cipherteks bertambah hingga 50%. Hal ini membuat metode ini kurang bagus jika dipakai untuk transfer data yang besar.

REFERENCES

[1] The Vigenere Cipher, http://www.cs.trincoll.edu/~crypto/historical/vigenere.html

[2] Cracking the Vigenere Cipher, http://www.vectorsite.net/ttcode_03.html#m2

[3] Slide Kuliah Kriptographi

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, 29 April 2010

Setia Negara B. Tjaru
13508054