Perancangan Cipher Baru untuk Huruf Korea (Hangul)

Nikodemus Adriel Limanthie/13510089
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
a.limanthie@yahoo.com

***Abstract*—Makalah ini membahas tentang ide penulis untuk membuat sebuah cipher baru untuk mengenkripsi huruf Korea yang biasa disebut *Hangul*. Pengenkripsian dilakukan dengan memanfaatkan komponen vokal dan konsonan yang ada pada *Hangul*. Komponen vokal dan konsonan tersebut akan dienkripsi dengan cara yang akan dijelaskan pada makalah ini agar menghasilkan suatu cipherteks berupa huruf korea pula.**

***Index Terms*—*Cipher*, *Hangul*, Komponen Konsonan, Komponen Vokal.**

I. PENDAHULUAN

Di dunia, kebutuhan pertukaran informasi pada zaman sekarang sangatlah besar. Segala macam hal, mulai dari komunikasi, hiburan, berita, hingga pendidikan membutuhkan adanya pertukaran informasi yang salah satu medianya adalah lewat media digital.

Pertukaran informasi lewat media digital sangat membantu perkembangan teknologi dan informasi manusia di seluruh dunia. Namun, pertukaran informasi lewat media digital bukan sesuatu yang aman karena mudahnya informasi tersebut didapatkan untuk digunakan pada hal yang salah. Enkripsi pesan dan informasi digunakan untuk mencegah hal tersebut terjadi.

Banyak algoritma enkripsi pada zaman sekarang yang menggunakan pengenkripsian pada level bit. Namun, masih ada algoritma enkripsi klasik yang menggunakan pengenkripsian per karakter, misalnya menggunakan vigenere cipher ataupun playfair cipher. Kekurangannya, algoritma enkripsi klasik tersebut tidak dapat mengenkripsi karakter-karakter di luar 256 karakter ASCII. Maka, untuk keperluan pengenkripsian karakter-karakter pada bahasa selain alfabet di dunia tidak dapat menggunakan algoritma enkripsi klasik yang sudah ada.

Oleh karena itu, pada pembahasan makalah saya kali ini, saya akan mencoba merancang sebuah algoritma enkripsi untuk bahasa Korea yang merupakan algoritma enkripsi klasik.

II. DASAR TEORI

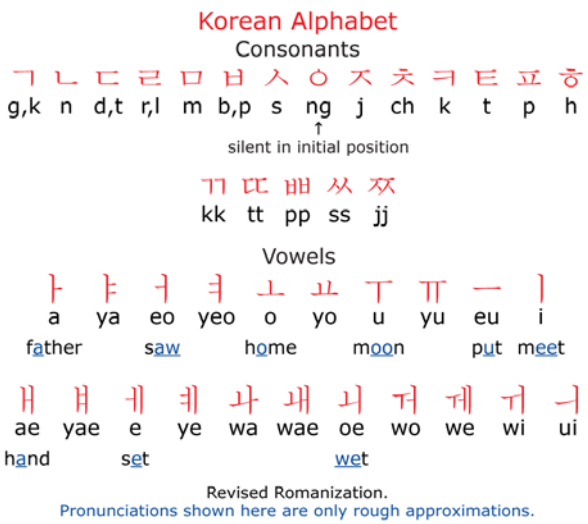
Algoritma kriptografi klasik adalah algoritma kriptografi yang melakukan pengenkripsian pesan secara karakter per karakter dan dapat dilakukan tanpa menggunakan alat-alat tradisional seperti kertas dan pensil. Algoritma kriptografi klasik pada umumnya sekarang sudah jarang digunakan karena mudahnya algoritma tersebut untuk dipecahkan dengan menggunakan komputer. Namun, algoritma kriptografi klasik merupakan dasar dari ilmu kriptografi pada zaman sekarang.

Algoritma kriptografi memiliki dua buah komponen, yaitu algoritma enkripsi/dekripsi dan kunci untuk enkripsi/dekripsi tersebut. Oleh karena adanya kunci tersebut, algoritma kriptografi tidak perlu disembunyikan terhadap orang lain karena yang memegang peran penting adalah kunci dari algoritma tersebut.

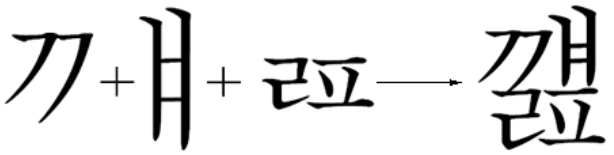
Algoritma kriptografi mengenkripsi berdasarkan karakter per karakter dari pesan. Pada bahasa Korea, kalimat atau kata dibentuk oleh karakter-karakter sehingga dapat diterapkan algoritma enkripsi klasik padanya. Perbedaan yang mungkin sedikit terlihat adalah karakter pada bahasa Korea bukan hanya terdiri dari satu huruf, namun merupakan gabungan huruf vokal dan konsonan.

Hangul pada mulanya diperkenalkan oleh King Sejong yang merupakan salah satu raja di Korea pada zaman dahulu. Pada waktu itu, Korea masih menggunakan bahasa Cina yang sangat sulit untuk dipelajari dan biasanya hanya dimengerti oleh bangsawan. Oleh karena itu, King Sejong memperkenalkan Hangul sebagai huruf yang mudah dipelajari agar rakyat jelata pun dapat membaca dan menulis. Dari situlah muncul Hangul yang digunakan sekarang.

Pada huruf Korea (Hangul) terdapat 21 huruf yang mewakili bunyi vokal dan 19 huruf yang mewakili bunyi konsonan. Huruf-huruf tersebut dapat digabungkan untuk membentuk sebuah karakter. Huruf-huruf tersebut dapat dilihat pada gambar berikut.



Pembentukan karakter baru dari gabungan huruf vokal dan konsonan dapat dilihat pada gambar berikut.



Seperti pada gambar berikut, huruf-huruf vokal dan konsonan digabung menjadi sebuah karakter. Penggabungan tersebut tidak sembarang diletakkan, namun terdapat aturan tertentu. Misalnya konsonan pertama akan diletakkan di kiri atas karakter. Lalu, dapat juga ditambahkan konsonan di akhir karakter yang diletakkan di bawah. Konsonan yang diletakkan di akhir karakter dapat berisi dua buah konsonan tertentu yang memang sudah disetujui oleh bahasa Korea untuk dapat digunakan, misalnya seperti pada contoh di atas, konsonan gabungan antara ㄹ(l) dan ㅍ(p) diletakkan di akhir.

Namun, untuk algoritma enkripsi yang saya gunakan, saya hanya akan menggunakan 14 huruf vokal dan 14 huruf konsonan tertentu. Alasannya, beberapa huruf pada Hangul dapat dibentuk dari 2 huruf lainnya sehingga dapat dianggap gabungan 2 huruf alih-alih satu huruf lain. Dengan samanya jumlah vokal dan konsonan pada algoritma enkripsi, maka saya menemukan sebuah algoritma enkripsi yang memanfaatkan hal tersebut yang akan saya jelaskan pada makalah ini.

III. IMPLEMENTASI DAN ANALISIS

Seperti yang telah saya sebutkan sebelumnya, saya akan menggunakan 14 huruf vokal dan 14 huruf konsonan yang menurut saya merupakan dasar dari semua huruf pada Hangul. Ke-14 huruf vokal dan konsonan tersebut adalah:

14 huruf vokal

ㅏ ㅐ ㅑ ㅒ ㅓ ㅔ ㅕ ㅖ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ

14 huruf konsonan

ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ

Sedangkan untuk huruf lainnya, dapat dilihat bahwa huruf-huruf tersebut merupakan gabungan dari 14 huruf dasar tadi. Huruf-huruf gabungan tersebut adalah seperti berikut.

ㄲ = ㄱ + ㄱ
ㄸ = ㄷ + ㄷ
ㅃ = ㅂ + ㅂ
ㅆ = ㅅ + ㅅ
ㅉ = ㅈ + ㅈ
ㅘ = ㅗ + ㅏ
ㅙ = ㅗ + ㅐ
ㅚ = ㅗ + ㅣ
ㅝ = ㅜ + ㅓ
ㅞ = ㅜ + ㅔ
ㅟ = ㅜ + ㅣ
ㅢ = ㅡ + ㅣ

Hangul pada umumnya dibentuk oleh gabungan konsonan dan vokal. Untuk melakukan enkripsi dengan cara yang saya buat, maka karakter-karakter pada sebuah kata atau kalimat harus dipecah menjadi huruf vokal dan konsonannya. Contoh:

한(han) = ㅎ(h) + ㅏ(a) + ㄴ(n)
글(geul) = ㄱ(g) + ㅡ(eu) + ㄹ(l)
떡(tteok) = ㄷ(d) +ㄷ(d) + ㅓ(eo) +ㄱ(g)
원(won) = ㅇ(-) +ㅜ(u) + ㅓ(eo) +ㄴ(n)

Dengan huruf-huruf yang didapatkan tersebut, maka dapat dilakukan enkripsi untuk huruf per huruf. Cara pengenkripsian dilakukan dengan memanfaatkan semacam rotor seperti pada enigma cipher. Terdapat dua buah rotor, yaitu rotor untuk huruf vokal dan rotor untuk huruf konsonan. Isi dari rotor tersebut berurutan sesuai dengan urutan abjad pada Hangul. Tampilan rotor tersebut seperti berikut.

Rotor untuk huruf vokal

ㅏ
ㅐ
ㅑ
ㅒ
ㅓ
ㅔ
ㅕ
ㅖ
ㅗ
ㅘ
ㅙ
ㅚ
ㅛ
ㅜ
ㅝ
ㅞ
ㅟ
ㅠ
ㅡ
ㅢ
ㅣ

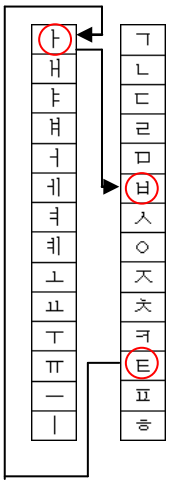
Rotor untuk huruf Konsonan

ㄱ
ㄴ
ㄷ
ㄹ
ㅁ
ㅂ
ㅅ
ㅇ
ㅈ
ㅊ
ㅋ
ㅌ
ㅍ
ㅎ

Lalu, untuk melakukan enkripsi tentu saja dibutuhkan kunci tertentu. Kunci pada enkripsi ini berupa bilangan minimal angka 1. Untuk contoh berikut, misalkan kita menggunakan kunci dengan angka 12 untuk mengenkripsi sebuah huruf ㅏ(a). Maka, proses pengenkripsian akan dilakukan seperti berikut.

Dicari huruf ㅏ(a) pada rotor (dilingkari merah)
Posisinya diberi pointer berupa lingkaran merah
Pointer tersebut diseberangkan ke rotor konsonan
lalu digeser sebanyak digit pertama kunci (1)
Setelah itu, Pointer diseberangkan lagi ke rotor vokal, dan digeser sebanyak digit ke-2 kunci (2)
Karena kunci hanya 2 digit, maka berakhirlah proses enkripsi dan didapatkan hasil enkripsi huruf ㅏ(a) yaitu huruf ㅒ(yae).

Karena urutan huruf tersebut bersifat sebagai rotor, maka akan terjadi pengulangan setelah huruf ㅣ(i) pada rotor vokal kembali ke huruf ㅏ(a), dan juga pengulangan setelah huruf ㅎ(h) pada rotor konsonan kembali ke huruf ㄱ(g). Sebagai contoh, misalkan akan dienkripsi sebuah huruf ㅌ(t) dengan kunci berupa angka 35. Proses pengenkripsian tersebut dapat dilihat seperti berikut.



Dicari huruf ㅌ(t) pada rotor (dilingkari merah)
Posisinya diberi pointer berupa lingkaran merah
Pointer tersebut diseberangkan ke rotor vokal
lalu digeser sebanyak digit pertama kunci (3)
Karena digit pertama kunci melebihi batas rotor maka pointer berulang kembali ke atas rotor.
Demikian dilanjutkan seperti biasa hingga menemukan hasil.

Jika algoritma enkripsi tersebut dituliskan dalam bahasa pemrograman, maka algoritma tersebut adalah seperti berikut.

```
List<Char> vokal;
List<Char> konsonan;

Char EncryptHangul (Char c, String key)
{
 Char charnow = c;
 String keynow = key;
 while (keynow != "")
 {
  if (vokal.contains(charnow))
  {
   int i = vokal.getIndex(charnow);
   i += Integer.parseInt(keynow.charAt(0));
   keynow.substring(1);
   if (i >= 14) i -= 14;
   charnow = konsonan[i];
  }
  else
  {
   int i = konsonan.getIndex(charnow);
   i += Integer.parseInt(keynow.charAt(0));
   keynow.substring(1);
   if (i >= 14) i -= 14;
   charnow = vokal[i];
  }
 }
 return charnow;
}
```

Dengan demikian, diperlukan juga sebuah algoritma untuk mendekripsi hasil yang sudah dienkripsi. Dengan cara pengenkripsian seperti itu, maka cara mendekripsinya hanya dengan dibalik urutan pengerjaannya. Kunci akan dibaca dari belakang dan iterasi dilakukan mundur alih-alih maju seperti pada enkripsi. Algoritma dekripsi adalah seperti berikut.

```
List<Char> vokal;
List<Char> konsonan;

Char DecryptHangul (Char c, String key)
{
 Char charnow = c;
```

```
 String keynow = key;
 while (keynow != “”)
 {
  if (vokal.contains(charnow))
  {
   int i = vokal.getIndex(charnow);
   int kl = keynow.length();
   i -= Integer.parseInt(keynow.charAt(kl-1));
   keynow.substring(0, kl-1);
   if (i < 0) i += 14;
   charnow = konsonan[i];
  }
  else
  {
   int i = konsonan.getIndex(charnow);
   int kl = keynow.length();
   i -= Integer.parseInt(keynow.charAt(kl-1));
   keynow.substring(0, kl-1);
   if (i < 0) i += 14;
   charnow = vokal[i];
  }
 }
 return charnow;
}
```

Algoritma tersebut bersifat seperti substitution cipher dimana sebuah huruf jika dienkripsi akan menghasilkan cipherteks yang sama. Hal ini disebabkan oleh penggunaan huruf yang sama dengan kunci yang sama akan melewati proses yang sama pula sehingga menghasilkan cipherteks yang sama. Walaupun digunakan kunci yang cukup panjang sehingga proses pengenkripsian menjadi sulit, hal tersebut akan tetap menjadi masalah.

Solusi dari masalah ini adalah dengan mengubah urutan rotor vokal atau konsonan tiap kali pengenkripsian, layaknya pada enigma cipher. Misalnya, ketika ditemukan hasil enkripsi pada huruf ㅑ(ya), maka huruf tersebut akan diubah menjadi huruf pertama dari rotor vokal sehingga mengubah urutan yang pertama tetap. Hal ini akan mengakibatkan pengenkripsian yang tidak mudah ditebak karena suatu huruf tidak selalu dienkripsi menjadi huruf yang sama.

Namun, pada makalah ini saya tidak membahas hal tersebut karena memerlukan pembahasan lebih mendalam.

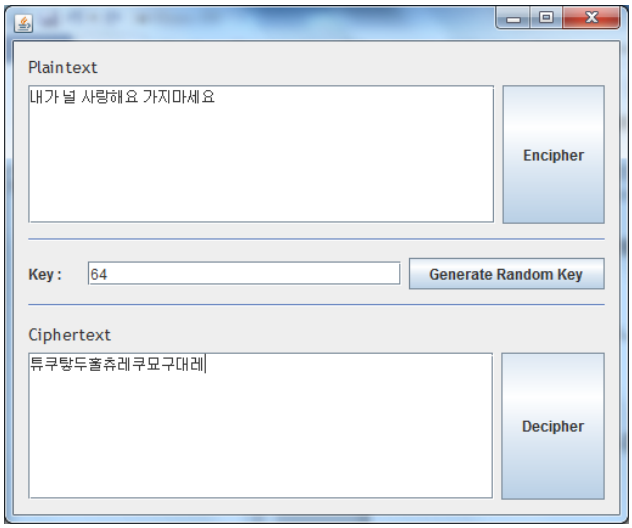
Algoritma tersebut saya terapkan pada sebuah program berbahasa Java yang mengaplikasikan cara yang sama. Misalkan digunakan sebuah kalimat berikut sebagai plainteks.

내가 널 사랑해요 가지마세요.

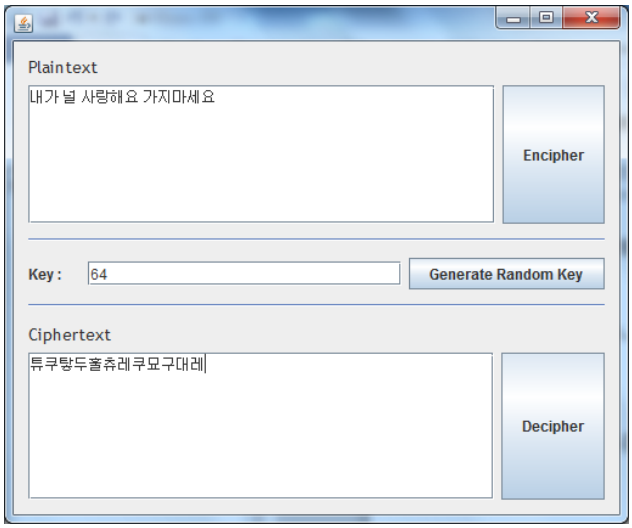
Jika dienkripsi menggunakan program yang saya buat dengan kunci berupa bilangan 64, maka hasil enkripsinya adalah seperti berikut.

튜쿠탛두훌츄레쿠묘구대레.

Kalimat tersebut menjadi tidak berarti setelah dienkripsi. Berikut ini adalah *screenshot* dari enkripsi pada program.



Untuk hasil ketika cipherteks di-decipher adalah seperti berikut.



Dengan demikian, algoritma cipher untuk huruf Korea yang saya buat telah berhasil.

IV. KESIMPULAN

Dari analisis tersebut, dapat diambil kesimpulan bahwa dapat pula dibentuk algoritma enkripsi untuk bahasa yang tidak menggunakan alfabet, salah satunya bahasa Korea. Dan walaupun algoritma yang sekarang masih belum cukup kuat, namun algoritma tersebut masih dapat dikembangkan agar dapat menjadi algoritma yang lebih kuat.

REFERENSI

[1] http://www.omniglot.com/writing/korean.htm.
[2] http://docs.oracle.com/javase/6/docs/api/java/util/List.html
[3] http://www.unicode.org/charts/PDF/UAC00.pdf.
[4] http://www.unicode.org/charts/PDF/U1100.pdf.
[5] http://users.telenet.be/d.rijmenants/en/handciphers.htm.
[6] Munir, Rinaldi. 2009. “Diktat Kuliah IF3058, Kriptografi,” Bandung : Program Studi Teknik Informatika ITB.

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

Nama dan NIM