PENYANDIAN NOMOR KARTU KREDIT UNTUK KEAMANAN TRANSAKSI ONLINE

Imaduddin Amin – NIM : 13505067

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15067@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang salah satu cara yang dapat dilakukan untuk meningkatkan keamanan dalam transaksi online yang menggunakan kartu kredit. Dalam beberapa transaksi online, untuk melakukan transaksi yang berbasiskan kartu kredit hanya dibutuhkan nomor dan masa berlaku kartu tersebut. Hal inilah yang dirasa tidak aman oleh sebagian orang karena validasi yang ada sangat minim dan mungkin didapatkan oleh orang lain. Untuk itulah diperlukan salah satu cara untuk menaggulanginya.

Kartu-kartu kredit yang ada sekarang berbasiskan pada algoritma cek Luhn, sehingga untuk mendapatkan nomor kartu kredit yang valid bukanlah sesuatu hal yang sukar untuk dilakukan. Seseorang bisa melakukan algoritma *brute-force attack* untuk mendapatkan nomor yang valid berdasarkan algoritma cek Luhn tersebut.

Dengan membuat *generate code* berbasiskan algoritma cek Luhn dan menggunakan *sosial engineering* seseorang bisa mendapatkan nomor kartu kredit yang valid sekaligus masa berlaku kartu tersebut. Tentu saja hal ini memungkinkan seseorang untuk menggunakan kartu yang bukan miliknya untuk dimanfaatkan. Oleh karena itu dibutuhkan cara lain guna mengurangi penyalah gunaan terhadap kartu.

Kata kunci: algoritma Luhn, *sosial engineering,* kartu kredit, enkripsi, PIN, *brute-force*, chiperteks, plainteks.

1. Pendahuluan

Kemajuan teknologi informasi dan tuntutan zaman yang serba instant menimbulkan satu tren baru dalam dunia perdagangan. Dahulu kala jual-beli terjadi dalam satu majelis tempat antara pejual dan pembeli namun di masa kini proses jual-beli dapat terjadi meskipun penjual dan pembeli tidak berada dalam satu majelis tempat. Pada akhirnya, sekarang ini – proses jual-beli dapat dilakukan dengan cara *on-line* berbasikan kartu kredit.

Kartu kredit adalah kartu yang diterbitkan oleh perusahaan yang memberikan jasa kredit dalam artian memberikan kemudahan dalam proses peminjaman uang untuk nasabahnya.

Setiap kartu kredit akan mempunyai nomor-nomor pengenal yang mewakili kartu tersebut. Nomor-nomor tersebut merupakan angka-angka yang tersusun berdasarkan algoritma Luhn.

Adapun cara untuk dapat bertransaksi berbasiskan kartu kredit ini, seseorang membutuhkan nomor kartu dan masa berlaku yang valid. Tentu saja hal ini sangat memudahkan dan efisien namun dari segi keamanan hal ini masih meragukan karena memungkinan orang lain untuk memanfaatkan kartu kredit yang bukan miliknya. Untuk itu diperlukan cara baru dalam proses jual-beli online ini guna mengurangi penyalah gunaan yang terjadi**.**

2. Kartu Kredit dan Algoritma Luhn

2.1 Kartu Kredit

Konsep penggunaan kartu dalam transaksi perbankan sebenarnya telah dikenal lebih dari 67 tahun yang lalu. Meski demikian, teknologi tinggi baru muncul sekitar dekade 1970-an. Pada masa ini muncul pertama kali mesin ATM yang menandai transaksi perbankan yang ditunjang

oleh teknologi telekomunikasi secara on line untuk semua nasabah selama 24 jam. Tiga puluh tahun kemudian, gaya transaksi elektronik ini menjadi gaya hidup lebih dari 90 persen transaksi perbankan di negara - negara maju.

Berikut ini sejarah perkembangan layanan kartu kredit yang ada di dunia :

1. Tahun 1924, Konsep penggunaan kartu dalam transaksi perbankan telah mulai diperkenalkan. Beberapa tahun kemudian metode pemakaian kartu ini diikuti oleh 100 buah bank di seluruh dunia.

2. Tahun 1950, Dinners Club dan American Express menjadi kartu yang menggunakan plastik pertama.

3. Tahun 1958, American Express menawarkan kartu untuk pasar travel dan entertainment.

4. Tahun 1966, Bank of Amerika menawarkan lisensi Kartu Amerika Bank ke bank - bank lain untuk membuat kartu pembayaran.

5. Tahun 1969, ATM (Automatic Teller Machine) pertama muncul di Inggris.

6. Tahun 1970, Ide pembuatan kartu kredit diterima secara luas.

7. Tahun 1977, Bank Americard memberi lisensi kartu kredit yang dipusatkan bersama secara resmi dibawah nama Visa.

8. Tahun 1995, Lebih dari 90 persen transaksi perbankan di Amerika dilakukan secara elektronik.

2.2 Algoritma Luhn

Nomor-nomor yang mewakili kartu kredit terdiri atas beberapa digit tergantung pada perusahaan yang mengeluarkannya, namun angka-angka tersebut semuanya berdasarkan pada algoritma cek Luhn.

Berikut ini daftar jumlah digit kartu berdasarkan perusahaan yang mengeluarkannya :

Panjang nomor	Jenis Kartu
15	JCB
14	Diners
15	Amex
16	JCB
13/16	Visa
16	MasterCard
16	Bankcard
16	Discover

Untuk dapat memastikan bahwa nomor yang tertera pada kartu kredit itu valid dapat dilakukan dengan 3 langkah yaitu :

- untuk setiap digit pada posisi yang ganjil kalikan nilainya dengan dua, jika hasilnya lebih dari 9, kurangi hasilnya dengan 9. Jumlahkan semua angka yang telah didapat itu.

- untuk setiap digit pada posisi genap, jumlahkan semua nilainya dan tambahkan hasilnya dengan hasil langkah pertama.

- jika hasil pada langkah kedua habis dibagi 10, berarti nomor kartu tersebut sah.

Langkah-langkah inilah yang disebut dengan algoritma Luhn.

Sebagai contoh, misalkan sebuah kartu memiliki nomor : **5541-9734-4563-1272**

Maka didapatkan digit-digit Ganjilnya adalah:

5	4	9	3	4	6	1	7

Sedangkan digit-digit Genapnya:

5	1	7	4	5	3	2	2

Untuk mengecek, keabsahan nomor kartu tersebut maka digunakan metode yang telah disebutkan diatas.

Langkah pertama :

5 x 2 = 10 (* Lebih dari 9 – kurangi dengan 9 *)
= 1
4 x 2 = 8
9 x 2 = 18 (* Lebih dari 9 – kurangi dengan 9 *)
= 9
3 x 2 = 6
4 x 2 = 8
6 x 2 = 12 (* Lebih dari 9 – kurangi dengan 9 *)
= 3

1 x 2 = 2
7 x 2 = 14 (* Lebih dari 9 – kurangi dengan 9 *)
= 5

Jumlah = 1+9+6+8+3+5 = 32

Langkah kedua :

Jumlah = 5 + 1 + 7 + 4 + 5 + 3 + 2 + 2 = 29

Langkah ketiga :

Jumlah = 32 + 29 = 63

63 mod 10 != 0;

Nomor tersebut tidak valid, karena tidak memenuhi algoritma Luhn, sehingga dapat dipastikan bahwa tidak ada kartu kredit yang memiliki seri pengenal **5541-9734-4563-1272.**

Sebagai contoh berikutnya, didapatkan bahwa Kartu lainnya memiliki nomor :

7889-8594-5435-5413

Maka digit-digit ganjilnya adalah

7	8	8	9	5	3	5	1

Langkah pertama :

7 x 2 = 14 (* Lebih dari 9 – kurangi dengan 9 *)
= 5
8 x 2 = 16 (* Lebih dari 9 – kurangi dengan 9 *)
= 7
8 x 2 = 16 (* Lebih dari 9 – kurangi dengan 9 *)
= 7
9 x 2 = 18 (* Lebih dari 9 – kurangi dengan 9 *)
= 9
5 x 2 = 10 (* Lebih dari 9 – kurangi dengan 9 *)
= 1
3 x 2 = 6
5 x 2 = 10 (* Lebih dari 9 – kurangi dengan 9 *)
= 1
1 x 2 = 2

Jumlah = 38

Langkah kedua :

Digit-digit Genap :

8	9	5	4	4	5	4	3

Jumlah = 42

Langkah ketiga :

Jumlah = 38 + 42 = 80

80 mod 10 == 0;

Berdasarkan cek Luhn maka **nomor tersebut valid.** Artinya nomor seri tersebut memungkinkan sebagai nomor seri dari sebuah kartu kredit.

Contoh di atas merupakan contoh pengecekan nomor kartu dengan jumlah digit genap. Untuk kartu yang jumlah digit nya ganjil, caranya sama, hanya saja pada langkah pertama yang dikalikan adalah digit pada posisi genap, dan pada langkah kedua yang dijumlahkan adalah digit pada posisi ganjil.

Contoh : Sebuah kartu memiliki nomor : **123406789101113** yang kita harus cek kebenaran nomornya. Maka berdasarkan algoritma cek Luhn untuk kartu kredit dengan jumlah digit ganjil. Maka
Langkah 1 :
Digit-digit genapnya adalah 2,4,6,8,1,1,1
2 * 2 = 4
4 * 2 = 8
6 * 2 = 12 (* Lebih dari 9 – kurangi dengan 9 *)
= 3
8 * 2 = 16 (* Lebih dari 9 – kurangi dengan 9 *)
= 7
1 * 2 = 2
1 * 2 = 2
1 * 2 = 2
Jumlah : 28

Langkah 2 :
Jumlah : 1 + 3 + 0 + 7 + 9+0+1+3 = 22

Langkah 3
Jumlah 22+28 = 50

50 mod 10 = 0

Kesimpulan, nomor tersebut valid.

2.2.1 Mendapatkan Nomor yang Valid

Berdasarkan algoritma luhn tersebut kita dapat mengenerate code untuk mendapatkan nomor yang valid. Kita hanya perlu membuat algoritmanya dalam bahasa pemrograman

tertentu, memasukkan input, mencek secara traversal dan menampilkan semua nomor yang valid.

Masukan yang memungkinkan adalah

00000000000000
00000000000001
…
…
99999999999999

Berdasarkan teori kombinarorial didapatkan bahwa masukan yang harus di test sebanyak (10^{16}) jika jumlah digit 16, Didapat dari

10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 * 10 = (10^{16})

10 = Jumlah element (0,1,2,3,4,5,6,7,8,9)

Untuk masukan sebesar ini tentu membutuhkan waktu komputasi yang sangat lama, jika dimisalkan komputer yang digunakan untuk mengenerate mampu menjalankan masukan berukuran n dalam waktu (10^{-4}) x (2^n) detik maka waktu yang dibutuhkan adalah (10^{-4}) x ($2^{10^{\wedge}\ 16}$) detik. Waktu yang sangat lama, oleh karena itu masukan harus dibatasi.

Berikut ini adalah contoh program untuk mengecek kebenaran nomor kartu kredit dan mnampu mencari nomor kartu yang benar yang terdekat dari nomor kartu yang salah dalam bahasa Javascript.

```
<script language="JavaScript"
type="text/javascript">

//menghilangkan spasi dan '-' baik di awal, di
tengah maupun di akhir string
function trimSpaces(s){
      var res;
      var i;
      res = "";
      for (i = 0; i < s.length; i++){
            if ((s.charAt(i) != " ") && (s.charAt(i)
!= "-"))
                  res += s.charAt(i);
      }
      return res;
}

//mengembalikan benar jika input yang
diberikan benar (semuanya angka)
//string dianggap telah dilewatkan ke trimSpaces
function isValidInput(s){
      for (i = 0; i < s.length; i++){
            var i;
            if ((s.charAt(i) < "0") || (s.charAt(i) >
"9"))
                  return false;
      }
      return true;
}

//membatasi angka agar tidak lebih dari 9
function fix(num){
      if (num <= 9) return num; else return (num
- 9);
}

//untuk mengecek kebenaran dengan 'luhn check
digit algorithm'
function check(number){
      var i;
      var ganjil;
      var genap;
      var tanda;

      genap = 0;
      ganjil = 0;
      //tanda = 1 artinya jumlah digitnya ganjil
      if (number.length % 2) tanda = 0; else
tanda = 1;
      for (i = 0; i < number.length; i++) {
            if ((i + tanda) % 2) //ganjil
                  ganjil += fix(2 *
(number.charAt(i)));
            else
                  genap +=
parseInt(number.charAt(i), 10);
      }
      return (((ganjil + genap) % 10) == 0);
}

//fungsi utama
function validateInput(inp){
      var tmp;
      var Msg;
      var Msg2;
      tmp = trimSpaces(inp.nomor.value)
      if ((tmp == "") || (!isValidInput(tmp))){
            alert("Data yang Anda masukkan
salah, baca keterangan");
            return false;
      }
      Msg = "Menurut Algoritma Check Digit
Luhn angka tersebut ";
      Msg2 = "sebagai angka kartu kredit";
      if (check(tmp))
```

```
        alert(Msg + "\n\nVALID\n\n" +
Msg2);
    else
        alert(Msg + "\n\nTIDAK VALID\n\n"
+ Msg2);
    return false;
}

//mencari beberapa angka valid yang dekat
dengan nomor yang diberikan
function findN(formName){
    var start;
    var startn;
    var res;
    var i;

    start =
trimSpaces(formName.nomor.value);
    if ((start == "") || (!isValidInput(start))){
        alert("Data yang Anda masukkan
salah, baca keterangan");
        return;
    }
    res = "Hasil :\n";
    startn = parseInt(start,10);
    for (i=-50; i<100; i++) {
        num = "" + (parseInt(start,10)+i);
        if (check(num)) {
            res += (startn + i) + "\n";
        }
    }
    formName.hasil.value = res;
}
//
//akhir skrip di sini
//
//-------------------------------------------------------->
</script>
```

2.3 Membatasi Masukan

2.3.1 Social Engineering

Sosial engineering adalah perform psikologis untuk mempengaruhi dan mencoba mengidentifikasi sebuah masalah atau dalam hal ini lawan. Ini adalah seni didalam dunia elektron.

Sebagai contoh, seseorang ingin mendapatkan nama-nama nasabah bank yang mempunyai rekening dengan saldo mininim 1 milyar. Sebagai tambahan, setiap nasabah yang memiliki rekening di atas 1 milyar setiap bulannya akan mendapat kue ucapan terimakasih dari pihak bank. Daripada melakukan pembobolan system keamanan bank tersebut untuk mendapatkan nama-nama nasabah yang mempunyai rekening di atas 1 milyar lebih baik mencari tahu, mana toko kue langganan bank kemudian mencari tahu siapa-siapa yang mendapat kiriman dari pihak bank.

2.3.1 Memanfaatkan Social Engineering

Untuk dapat membatasi masukan yang besar, seseorang bisa menggunakan teknik social engineering . Hal ini mungkin dilakukan dengan cara membuat kartu kredit di bank tertentu untuk mendapatkan nomor kartu lainnya yang berdekatan dengan nomor kartu yang dia dapat.

Nomor kartu yang dikeluarkan oleh pihak yang berhak secara bersamaan kemungkinan memiliki masa berlaku yang sama.

Sebagai contoh, kartu kredit yang dikeluarkan oleh pihak produsen untuk sekelompok mahasiswa di salah satu perguruan tinggi atau sejumlah karyawan disuatu perusahaan memiliki jumlah digit 16 dan memiliki masa berlaku yang sama. Namun untuk setiap kartu, 12 digit pertamanya selalu diawali dengan 5265 2323 6113 2946 dengan 4 digit random yang mengikuti. Oleh karena itu, tentu untuk mendapatkan nomor kartu sekelompok mahasiswa atau pekerja tersebut seseorang hanya perlu melakukan generate code dengan masukan 5265 2323 6113 2946 0000 sampai dengan 5265 2323 6113 2946 9999 hal ini akan mempercepat proses komputasi karena kemungkinan masukannya sekarang tinggal 10 x 10 x 10 x 10 = 10^4

3. Ketidak Amanan Transaksi Berbasis Kartu Kredit

Seperti telah dipaparkan pada bagian pendahuluan, meskipun mudah dan efisien, penggunaan kartu kredit dalam pembayaran transaksi online masih tergolon sangat riskan.

Sebagai contoh kasus, sebuah LSM memberi kesempatan kepada masyarakat untuk dapat beramal. Agar mempermudah masyarakat dalam beramal, LSM tersebut memberikan kesempatan kepada masyarakat untuk menyalurkan amalnya melalui SMS dengan cara mengetikkan

"REG<spasi>nomor_kartu_kredit <spasi>masa_berlaku_kartu<spasi>jumlah_uang#".

Hal ini dirasa sangat membantu karena orang yang ingin beramal tidak perlu repot untuk mentransfer uang ataupun datang langsung ke loket pembayaran.

Namun bisa saja ada orang lain yang mengatasnamakan seseorang untuk transaksi tersebut dengan cara yang telah dijelaskan pada bagian 2 makalah ini. Untuk itulah diperlukan suatu metode baru dalam penggunaan kartu kredit sebagai alat pembayaran dalam transaksi online.

4. Penyandian Nomor Kartu

Untuk menghindari terjadinya penyalahgunaan kartu oleh orang lain. Maka dibutuhkan data tambahan selain nomor kartu dan masa berlaku kartu. Data tersebut haruslah hanya diketahui oleh pemegang kartu dan pihak yang mengeluarkan kartu tersebut. Salah satu yang memenuhi adalah PIN (*Personal Identification Numbers*). Namun PIN ini tidak boleh diketahui oleh pihak lain selain pemegang kartu dan pihak yang mengeluarkan kartu.

Alternatif yang bisa dipakai adalah PIN ini digunakan sebagai kunci untuk melakukan enkripsi.

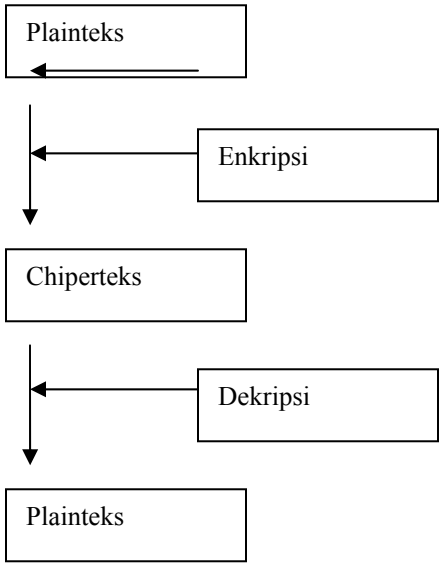
Cara seperti ini memang merepotkan karena pemilik kartu kredit harus membangkitkan nomor-nomor yang telah dienkripsi sebelumnya. Proses penyandian ini bisa dialih tugaskan kepada pihak penyedia kartu dengan cara mengirim nomor kartu dan PIN kartu tersebut.

Hal ini merupakan salah satu konsekuensi karena keamanan berbanding terbalik dengan kenyamanan.

4.1 Algoritma Enkripsi

Sebelum membahas mengenai enkripsi, tentu kita harus memahami apa yang dimaksud dengan Plainteks dan Chiper Text. Secara bahasa Plainteks berarti teks jelas yang dapat dimengerti sedangkan Chiperteks adalah teks yang tersandi. Sebagai contoh, sebuah pesan rahasia sebagai berikut : ” Aku Sayang Kamu ” disandikan menjadi chiperteks menjadi ”25588 777729992664 552688”.

Orang yang membaca plainteks maka dia akan langsung mengerti maksud dari isi tersebut sedangkan orang yang bisa membaca maksud dari chiperteks tersebut hanyalah orang yang mengetahui kunci dari penyandian tersebut. Sedangkan yang dimaksud dengan enkripsi adalah proses penyandian dari plainteks ke chiperteks.



Bagan Proses Enkripsi- Dekripsi

Contoh lainnya adalah sebuah data berisikan ” PESAN SANGAT RAHASIA” mengalami proses penyandian sehingga chiperteksnya adalah ” PHHW PH DIWHU WKH WRJD SDUWB”

Teknik penyandian tersebut menggunakan salah satu teknik lama yaitu *caesar chipper*. Tiap huruf pada plainteks disubtitusikan pada ketiga huruf berikutnya.

Plainteks : A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
Chiperteks : D E F G H I J K L M N O
P Q R S T U V W X Y Z

Dengan mengkodekan A sebagai 0 ... Z sebagai 25 maka rumus penyandiannya dapat dipetakan sebagai :

$Ci = E(pi) = (pi+3) \bmod 26$;

Untuk mengembalikan menjadi plainteks semula maka secara matematis persamaan

$Pi = D(ci) = (ci-3) \bmod 26$;

Misal huruf ”E” pada kata Pesan, E mempunyai indeks 4 pada jajaran alfabetis akan diubah menjadi bentuk chiperteks Ci = (4+3) mod 26 = 7 . Sehingga dalam bentuk chipertexnya menjadi H.

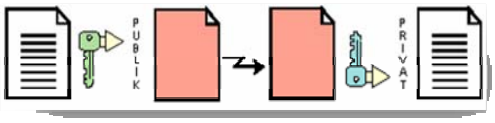
Sebaliknya untuk mengembalikan penyandian dari B. Dengan rumus Pi = (1-3) mod 26 = 24 sehingga didapatkan H menjadi E.

Selain contoh diatas, dikenal pula enkripsi yang mempunyai pola yang sama dengan *caesar chipper* diatas hanya saja, setiap indeks nya digeser dengan 13. Enkripsi ini dikenal dengan istilah ROT13 yang umum digunakan pada sistem unix. Berikut adalah contoh program yang menyandikan data dengan algoritma ROT13 dalam bahasa PERL yang dibuat oleh Budi Raharjo

```
#! /usr/bin/perl
# rot13: rotate 13
# usageL rot13 < filename.txt
# bugs: only works with lower case
#
# Copyright 1998, Budi Rahardjo
#                 <rahard@paume.itb.ac.id>,
<budi@vlsi.itb.ac.id>
# Electrical Engineering
# Institut Teknologi Bandung (ITB), Indonesia
#
while (<>) {
# read a line into $_
for ($i=0 ; $i < length($_) ; $i++) {
$ch = substr($_,$i,1);
# only process if it's within a-z
# otherwise skip
if ( (ord($ch)>=97) and (ord($ch)<=122) ) {
$newch = &rot13($ch); # rotate it
printf("%c", $newch);
} else {
# just print character that was not processed
print $ch;
}
} # end for loop
} # done...
sub rot13 {
local($ch) = @_;
$asch = ord($ch) - 97; # get the ascii value and
normalize it
$rotasch = $asch + 13; # rotate 13 it
# send it back to ascii
$rotasch = $rotasch % 26;
$rotasch = $rotasch + 97;
return($rotasch);
}
```

Karena kunci dekripsi sama dengan kunci enkripsi maka algoritma enkripsi tersebut disebut juga algoritma simetri. Jika kunci dekripsi tidak sama dengan kunci enkripsi maka algoritmanya disebut algoritma ninsimetri.

Contoh dari algoritma enkripsi ninsimetri ini adalah algoritma RSA. RSA adalah algoritma yang diperkenalkan oleh (Riverst-Shamir-Adleman). Dalam algoritma ninsimetri ini dikenal istilah kunci publik Kunci publik adalah kunci enkripsi yang sifatnya diketahui oleh umum, kunci inilah yang dipergunakan dalam proses penyandian dari plainteks ke chiperteks. Sedangkan untuk kunci dekripsi, hanya diketahui oleh pihak yang berwenang jadi sifatnya adalah rahasia.



Gambar Kunci ninsimetris

Adapun langkah-langkah dari Algoritma RSA adalah sebagai berikut :

1. Pilih a dan b, keduanya bilangan prima dan harus dijaga kerahasiannya.
2. Hitung n = a x b. Nilai n tidak dirahasiakan.
3. Hitung m = (a – 1) x (b – 1). Setelah m didapat maka a dan b dihapus (untuk mencegah orang lain tahu)
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya e, yang relatif prima terhadap m.
5. Dapatkan kunci dekripsi, d, dengan kekongruenan ed ≡ 1 (mod m). Isi pesan dienkripsi dengan persamaan ci = pi^e mod n, yang dalam hal ini pi adalah blok plainteks (misal angka dengan indeks ke i) , ci adalah chiperteks yang diperoleh, dan e adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai pi harus terletak dalam himpunan nilai 0, 1, 2, …, n – 1 untuk menjamin hasil perhitungan tidak berada di luar himpunan.
6. Proses dekripsi dilakukan dengan menggunakan persamaan pi = cid mod n, yang dalam hal ini d adalah kunci dekripsi.

Contoh (dari contoh 21 diktat kuliah Matematika Dikrit Informatika ITB) :

Misalkan a = 47 dan b = 71 (keduanya prima), maka dapat dihitung
n = a x b = 3337 dan m = (a – 1)x(b – 1) = 3220.

Pilih kunci publik e = 79 (yang relatif prima dengan 3220). Nilai e dan m dapat dipublikasikan ke umum. Selanjutnya akan dihitung kunci dekripsi d seperti yang dituliskan pada langkah instruksi 4,

$$ed \equiv 1 \pmod{m}.$$

Persamaan tersebut dapat dirubah menjadi bentuk

$$d = \frac{1 + (k \times 3220)}{79}$$

sehingga kunci dekripsi dengan mencoba nilai-nilai k = 1, 2, 3, …, diperoleh nilai d yang bulat adalah 1019. Ini adalah kunci dekripsi.

Misalkan plainteks

P = HARI INI

atau dalam desimal ASCII:

7265827332737873

Pecah P menjadi blok yang lebih kecil (misal 3 digit):

p1 = 726	p4 = 273
p2 = 582	p5 = 787
p3 = 733	p6 = 003

Blok pertama dienkripsikan sebagai 72679 mod 3337 = 215 = c1.

Blok kedua dienkripsikan sebagai 58279 mod 3337 = 776 = c2.

Dengan melakukan proses yang sama untuk sisa blok lainnya, dihasilkan chiperteks C = 215 776 1743 933 1731 158.

Proses dekripsi dilakukan dengan menggunakan kunci rahasia d = 1019.

Blok c1 didekripsikan sebagai 2151019 mod 3337 = 726 = p1,

Blok c2 didekripsikan sebagai 7761019 mod 3337 = 582 = p2.

Blok plainteks yang lain dikembalikan dengan cara yang serupa. Akhirnya kita memperoleh kembali plainteks semula P = 7265827332737873 yang karakternya adalah P = HARI INI.

Perhitungan perpangkatan pada proses enkripsi (ci = pie mod n)dan dekripsi (pi = cid mod n) membutuhkan bilangan yang sangat besar. Untuk menghindari penggunaan bilangan yang besar, maka dapat digunakan penyederhanaan dengan persamaan berikut:

ab mod m = [(a mod m)(b mod m)] mod m

Dalam penggunaan Algoritma RSA ini tidak perlu meragukan kekuatannya dalam menyandikan data selama belum ada algoritma matematika untuk memfaktorkan bilangan non prima menjadi bilangan prima secara efisien.

4.1.1 Perbandingan Algoritma Enkripsi Simetri dan ninsimetri

Perbedaan prinsip dan penggunaan kriptografi kunci publik dan algorima enkripsi simetri merupakan satu kajian yang sangat luas. Akan tetapi pada dasarnya, seperti telah dijelaskan sebelumnya, algoritma enkripsi simetri, mempunyai satu kunci yang sama digunakan untuk melakukan enkripsi dan dekripsi. Pada sistem kunci publik , enkripsi dan dekripsi menggunakan kunci yang berbeda.

Sejak dikembangkannya algoritma penyandian kunci-publik, selalu timbul pertanyaan mana yang lebih baik. Para pakar kriptografi mengatakan bahwa keduanya tidak dapat dibandingkan karena mereka memecahkan masalah dalam domain yang berbeda. Algoritma enkripsi simetri merupakan hal yang baik untuk mengenkripsi data. Dengan kelebihan dalam proses kecepatan komputasinya. Sementara algoritma kunci publik (ninsimetri) dapat melakukan hal-hal lain lebih baik daripada algoritma simetri, misalnya dalam hal key management.

4.2 Algoritma Simetri sebagai Algoritma penyandian nomor kartu

Dari penjelasan mengenai kedua bentuk dari algoritma enkripsi tersebut, penulis lebih

cenderung untuk memilih (dalam kasus ini) karena yang dibutuhkan dalam proses penyandian ini adalah satu validasi baru yang hanya diketahui oleh pihak bank dan pihak pribadi yang memiliki kartu kredit. Jika menggunakan algoritma ninsimetri maka akan ada kunci publik artinya setiap orang bisa menyandikan, tentu hal ini tidak bermanfaat dalam kasus ini. Namun jika menggunakan algortima penyandian simetri maka hanya dua pihak yang berkepentingan saja yang mengetahuinya.

4.3 Penerapan PIN sebagai kunci Private

Untuk penambahan validasi penulis lebih cenderung menggunakan PIN sebagai kunci private yang digunakan untuk kunci deskripsi maupun enkripsi terhadap nomor kartu. Karena PIN harus dijaga kerahasiannya dan sangat riskan apabila PIN tersebut yang dienkripsi untuk dijadikan validasi. Untuk itulah penulis rasa lebih tepat jika PIN hanya digunakan sebagai kunci enkripsi-dekripsi nomor kartu bukan media yang di enkripsi untuk dijadikan validasi.

Jika PIN yang di enkripsi kemudian dijadikan validasi maka akan ada kemungkinan seseorang (misal X) dapat membaca PIN yang sebenarnya. Misalkan PIN X adalah 7986. Setelah di enkripsi menjadi 1320. X ingin mengetahui PIN asli dari orang lain (Y) yang chiperteksnya adalah 2543. Dengan mempelajari chiperteks dan plainteks dari PIN yang dia (X) miliki, X dapat mengetahui bahwa chiperteksnya adalah setiap digit (dengan indeks i) mod angka yang paling kecil. Sehingga dengan chiperteks PIN Y yang diketahui, dia dapat mengetahui PIN asli dari Y tersebut. Dalam contoh diperlihatkan bahwa chiperteks dari Y adalah 2131 artinya, nilai paling kecil dari PIN Y adalah 6,7,8,9 (berdasarkan teorima modulo) dengan memasukkan 6 sebagai digit terkecil maka didapatkan kemungkinan PIN asli dari Y adalah 8797. Jika dimasukkan dengan angka 7,8 atau 9 maka tidak terdapat kemungkinan yang cocok karena tidak ada angka diatas 7,8,9 antara 1-9 yang dibagi 7 sisanya adalah 3 (digit ketiga dari chiperteks). Maka jelaslah jika PIN yang di enkripsi untuk dijadikan validasi maka masih terdapat banyak cara yang tidak begitu rumit untuk mengetahui PIN asli sebenarnya.

Karena itulah PIN yang dimiliki pemilik kartu hanya dipakai sebagai kunci bukan media enkripsi dalam proses ini. Namun meskipun PIN hanya dijadikan kunci enkripsi-dekripsi sebelumnya PIN harus disandikan terlebih dahulu juga agar tidak ada orang lain yang dengan memanfaatkan cara seperti seblumnya untuk mendapatkan kunci atau dengan kata lain PIN., Maka kunci yang digunakan untuk menyandikan nomor kartu adalah kunci yang merupakan hasil pengolahan dari PIN.

4.4 Penyandian nomor Kartu

Pada bagian sebelumnya telah dijelaskan mengapa nomor kartu kredit ini harus disandikan terlebih dahulu untuk melakukan transaksi sekaligus telah dijelaskan pula mengapa kunci dari penyandian adalah hasil penyandikan PIN sebelumnya.

Untuk itu langkah yang harus dilakukan terlebih dahulu sebelum penyandian nomor kartu ini adalah penyandian PIN nya terlebih dahulu. Sebagai alternatif dalam penyandian PIN ini kita bisa menggunakan sembarang algoritma misalnya kita akan menyandikan PIN dengan rumus ci = pi mod n. Dengan n adalah salah satu dari digit-digit PIN yang ada.

Misal PIN sebuah kartu adalah 7866 dan kita memilih n = 6 maka hasil penyandiaanya adalah 7 mod 6, 8 mod 6, 6 mod 6, 6 mod 6 sehingga chiperteks nya adalah 1200. Dengan memanfaatkan hasil yang diperoleh dari penyandian PIN sebelumnya maka penyandian nomor kartu dapat dilakukan, dengan menggunakan hasil penyandian PIN sebagai kunci penyandian nomor kartu.

Misal nomor kartu yang ingin disandikan adalah **7889-8594-5435-5413** dengan PIN 7869. Jika rumus penyandian PIN adalah ci = pi mod n. Dengan n adalah 6. maka hasil penyandian dari PIN adalah 1200. Hasil ini digunakan untuk menyandikan nomor kartu. Misal nomor kartu kita bagi menjadi 4 blok dan rumus penyandian untuk tiap blok adalah ci=pi mod 1200 (1200 = hasil dari penyandian PIN) maka dengan contoh nomor kartu diatas didapatkan bahwa hasil penyandiaanya(chiperteksnya) adalah 7889 mod 1200, 8594 mod 1200, 5435 mod 1200, 5413 mod 1200 = 689, 194, 635, 613 sehingga hasil penyandian dari nomor kartu adalah **689194635613** dengan asumsi rumus penyandian adalah seperti yang dikerjakan pada contoh.

5. Keamanan Setelah Penyandian Nomor Kartu

Setelah mengalami proses penyandian, maka terdapat validasi baru dalam dunia transaksi online berbasiskan kartu kredit sehingga lebih aman. Penyandian ini sulit untuk dikembalikan karena menggunakan PIN yang hanya diketahui oleh pihak pemilik kartu dan penyedia kartu. Selain itu PIN yang harus dijaga kerahasiaannya tetap terjaga karena sebelum dijadikan sebagai kunci enkripsi PIN ini telah dirubah dulu ke bentuk lain (disandikan terlebih dahulu).
Jika kita menggunakan algoritma yang telah digunakan pada bagian 4.4 maka seorang penyusup yang ingin melakukan pencurian identitas kartu harus melakukan enkripsi terhadap nomor kartu kredit dengan kunci PIN yang telah disandikan. Peluang untuk bisa melakukan hal ini sangatlah kecil, sehingga kemungkinan besar tidak ada yang mampu untuk melakukan pencurian identitas kartu.

Namun apabila penyusup tersebut menggunakan kaidah *brute-force attact* si penyusup harus melakukan komputasi yang sangat lama (akan dijelaskan kemudian).

Dari bagian 4.4 jika kita melakukan penyandian dengan rumus ci=pi mod (sembarangdigitPIN) maka akan terdapat ribuan kemungkinan untuk chiperteks yang dihasilkan.

Jika jumlah digit PIN adalah 4 maka chiperteksnya akan mempunyai kemungkinan 10^4 kemudian dari kemungkinan ini akan didapatkan kunci penyandian nomor kartu sebanyak kombinasi tersebut. Waktu komputasi yang dibutuhkan untuk dapat melaksanakan proses coba-coba akan sangat lama karena peyusup harus melakukan enkripsi nomor kartu kredit yang ada dengan kunci sebanyak 10^4 .

6. Kesimpulan

Kesimpulan yang dapat diambil dari makalah ini adalah :

1. Nomor kartu kredit yang ada didasarkan pada algoritma Luhn sehingga bisa ditentukan dan dicari mana nomor kartu yang valid dan mana yang tidak
2. Transaksi online yang berbasiskan kartu kredit dengan hanya menggunakan validasi berupa nomor kartu, masa kadaluarsa kartu, nama pemilik masih tidak aman karena memungkinkan seseorang untuk emnyalahgunakan kartu orang lain dengan.
3. Sebaiknya untuk transaksi online berbasiskan kartu kredit ditambah validasi yang sifatnya hanya diketahui oleh pemilik kartu dan pihak penyedia layanan kartu dalam hal ini adalah nomor PIN.
4. Karena nomor PIN bersifat rahasia maka nomor PIN tidak digunakan sebagai media untuk validasi tambahan tetapi merupakan cara untuk mendapat validasi tambahan (dalam makalah ini digunakan sebagi kunci penyandian nomor kartu)
5. Metode penyandian dalam penyandian nomor kartu kredit ini menggunakan kunci privat (algoritma enkripsi simetri) karena hanya dua pihak yang harusnya mengetahui kunci (PIN) ini.
6. Validasi tambahan yang digunakan untuk bertransaksi yang sifatnya hanya diketahui oleh pemilik kartu dan pihak penyedia layanan kartu membuat transaksi lebih aman.

DAFTAR PUSTAKA

[1] Black April, Sejarah Layanan Kartu Kredit Dunia, *www.yogyakarding.com* Tanggal akses tahun 2003

[2] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[3] Pfleeger, Charles P. (1997). Security in Computing Second Edition. Prentice-hall Intenational, inc.

[4] Raharjo, Budi, (2002). Keamanan Sistem Informasi Berbasis Internet PT Insan Infonesia - Bandung & PT INDOCISC – Jakarta

[5] Rosen, Kenneth H., *Diskcrete Mathematics and Its Applications*, 4th, McGraw-Hill International 1994

[6] scut (Kecoak Elektronik), Social Engineering? Fenomena hacking psikologis (Edisi pemula), *www.k-elektronik.com*. Tanggal akses tahun 2003

[7] Yohanes, Algoritma Cek Karu Kredit, *www.klik-kanan.com*. Tanggal akses 29 Desember 2006 pukul 21.00