Perancangan Aplikasi Kriptografi Asimetris Dengan Menerapkan Metode Elliptic Curve Cryptography

Eka Indah Sari
STMIK Budi Darma Medan, Jl. SM.Raja No.338 Sp.Limun Medan, Sumut, Indonesia.
Email :ekaindahsari789@yahoo.com

ABSTRACT
Cryptography is one of the sciences used to maintain confidentiality and data security has been developing since ancient Greece. One cryptographic method that is quite reliable, stable and the parent of the popular cryptographic algorithms today is Elliptic Curve Cryptography (ECC). In this case the previous block encryption results are feedback into current block encryption. The trick, the current plaintext block is XORed first with the ciphertext block from the previous encryption, then the XOR results are entered into the encryption function. With ECC mode, each ciphertext block depends not only on the plaintext block but also on the all previous plaintext blocks, Implementation of the Elliptic Curve Cryptography (ECC) algorithm in this study was carried out by using an application built using visual basic programming language. Net 2008..

Kata Kunci : Kriptografi, Elliptic Curve Cryptography (ECC), Keamanan Teks.

PENDAHULUAN

Keamanan data merupakan salah satu bagian yang penting pada saat ini.Terutama pada pengguna komputer yang sering menyimpan data yang dimiliki pengguna tanpa terlebih dahulu mempassword ataupun menyembunyikan data tersebut. Salah satu aspek keamanan dalam dokumen konvensional dan digital adalah keaslian. Seperti dokumen *konvensional,* dokumen digital pun harus terjamin keasliannya, bentuk dan isinya harus sesuai dengan yang dimaksud oleh pembuat.

Pengiriman data atau informasi tanpa dilakukan pengamanan akan beresiko terhadap penyadapan [1]. Untuk mengamankan data-data/informasi tersebut dibutuhkan suatu metode yang dapat membuat data tersebut tidak dapat diketahui oleh orang lain selain orang yang dituju untuk menerima data yang dimaksud. Metode tersebut harus mampu untuk mengamankan data.Untuk melindungi akses data dari pihak-pihak yang tidak berkepentingan tersebut [2] maka sangat diperlukan enkripsi dan dekripsi.Agar dapat dilakukan dengan baik, dibutuhkan suatu algoritma untuk enkripsi dan dekripsi.Algoritma/metode yang digunakan adalah *Elliptic Curve Cryptography* (ECC).

Elliptic Curve Cryptography adalah kriptografi kunci publik. Pada kriptografi kunci publik, masing-masing *user* atau *device* mengambil bagian dalam komunikasi yang memiliki pasangan kunci yaitu kunci publik dan kunci *private* [3]. Hanya pengguna yang cocok yang dapat menggunakan *private key* yang sesuai, tetapi kunci public yang digunakan disebarkan kepada pihak yang akan mengirimkan data. Dipilihnya ECC *Elliptic Curve Cryptography* sebagai metode kriptografi untuk proteksi dokumen berdasarkan pada hal-hal berikut:

1. Besarnya *field* dimana kurva elips berada dapat dipilih sehingga memudahkan implementasi *Elliptic Curve Cryptography* pada suatu batasan tertentu.
2. Besar kunci yang dihasilkan dengan metode *Elliptic Curve Cryptography* tidak terlalu besar sehingga tidak membutuhkan banyak memori tambahan.
3. Proses kriptografi *Elliptic Curve Cryptography* tidak membutuhkan prosesor khusus sehingga bisa mengurangi biaya implementasi.

Agar pembahasan tidak keluar dari pokok bahasan,penulis memberikan batasan-batasan sebagai berikut : Yang dibahas dalam aplikasi kriptografi ini adalah pengamanan data berupa text seperti huruf [10].

LANDASAN TEORI

2.1 Kriptografi

Cryptography (kriptografi) berasal dari bahasa Yunani yaitu dari kata crypto yang berarti penulisan *secret* (rahasia), sedangkan graphein artinya *writing* (tulisan). Jadi secara sederhana dapat diartikan *secret writing* (tulisan rahasia). Definisi lain dari kriptografi adalah sebuah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti

kerahasiaan, integritas data serta otentikasi. Menurut terminologinya, kriptogrofi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. (Doni Ariyus, Pengantar Ilmu Kriptografi, 2006)

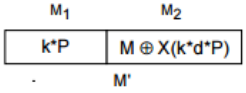
2.2 Elliptic Curve Cryptography

The Elliptic Curce Cryptosystem (ECC) diperkenalkan pada tahun 1985 oleh Neal Koblitz dan Victor Miller dari Universitas Washington. Kurva eliptik mempunyai masalah logaritma yang terpisah sehingga sulit untuk dipecahkan.Kriptografi kurva eliptik termasuk sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva eliptik.

The Elliptic Curce Cryptosystem (ECC) diperkenalkan pada tahun 1985 oleh Neal Koblitz dan Victor Miller dari Universitas Washington[7]. Kurva eliptik mempunyai masalah logaritma yang terpisah sehingga sulit untuk dipecahkan.Kriptografi kurva eliptik termasuk sistem kriptografi kunci publik yang mendasarkan keamanannya. Dalam proses enkripsi, pertama-tama dilakukan pembacaan suatu berkaskunci publik yang berisi kurva eliptik E, suatu point P yang berada pada E, suatu bilangan prima p ∈ Fp, dan kunci publik pemakai lain Q =d*P.Kemudian dipilih suatu bilangan random k ∈ {2,…,p-1} yang berubah untuk setiap blok data, dan dihitung k*Q dan k*P, selanjutnya berkas data dibaca secara per blok (M) dan dienkripsi dengan cara :

$$M' = [k*P, M \oplus X(k*Q)]$$

Keterangan :
M = data yang akan dienkripsi (plaintext).
M' = blok data yang telah dienkripsi (ciphertext).
k = suatu bilangan random yang akan digunakan sebagai session key dengan k ∈ {2,…,p-1}
Q = d*P
P = suatu point pada kurva E(Fp)
X(k*Q) = koordinat X untuk point yang dihasilkan dari perkalian k*Q.
M di-xor-kan dengan absis point yaitu k*Q, hasilnya berupa string yang lalu ditulis ke berkas dengan k*P ditambahkan sebelumnya. Hasil akhirnya secara sederhana dapat dilihat sebagai berikut :



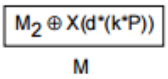
Keterangan :
M1 : session key.
M2 : Data terenkripsi.
M : Data yang belum terenkripsi.
M' : Blok data yang telah dienkripsi (ciphertext).

Proses ini akan terus dilakukan selama data yang dibaca masih ada.

Dalam proses dekripsi, pertama-tama dilakukan pembacaan suatu berkas kunci publik yang berisi kurva eliptik E, suatu point P yang berada pada E, dan suatu field bilangan prima p. Kemudian dibaca ciphertext seperti pada Gambar. 2.4. Lalu dihitung d*(k*P), dengan d adalah kunci privat yang dimasukkan oleh pemakai dalam bentuk passphrase, dan k*P berasal dari ciphertext. Satu buah blok data lalu dibaca (M'). Setelah itu dilakukan proses dekripsi untuk memperoleh M , dengan

$$M = [M2 \oplus X(d*(k*P))]$$

M2 di-xor-kan dengan absis point yaitu d*(k*P) sehingga diperoleh suatu string. Hasilnya (M) lalu ditulis ke berkas. Hasil akhirnya secara sederhana dapat dilihat sebagai berikut :



Keterangan :
M2 : Data terenkripsi.
M : Blok data yang telah didekripsi (plaintext).
Proses ini akan terus dilakukan selama data terenkripsi yang dibaca masih ada.

PEMBAHASAN

Pada proses enkripsi dan depenelitian dengan metode ECC adapun yang menjadi input adalah kunci rahasia (k) dan titik kurva (x,y). Titik kurva didapatkan dengan menggunakan rumus dan juga secara manual dengan memperhatikan titik x dan y. Sebelum melakukan proses enkripsi dan depenelitian terlebih dahulu ditentukan kunci umum (kP) dan kunci rahasia (kP).

Kunci publik (kP) akan dikirimkan bersama dengan pesan terenkripsi sebagai header yang akan dihitung oleh penerima pesan sebagai titik awal pendepenelitiansan pesan.

3.1. Tahap-Tahap Enkripsi

a. Tentukan plaintext
b. Tentukan kunci publik dengan rumus k*(x,y)
   Kunci rahasia (k)
   Titik kurva (x,y)
c. Proses
   k*(x,y) = kunci publik (kP)
   Menentukan titik enkripsi k*kP untuk mendapatkan titik absis x.
   Pesan diubah dan di xor kan dengan titik x yang menghasilkan pesan terenkripsi.
   Pesan dikirim dengan (kP sebagai header, pesan terenkripsi).
d. Output
   Pesan terenkripsi/*chipertext*

3.2. Tahap-Tahap Deskripsi

1. Ambil pesan terenkripsi
2. Masukkan kunci rahasia
   Kunci rahasia (k)
   Titik kurva (x,y) didapatkan dari header pesan terenkripsi sebagai kP
3. Proses
   Menentukan titik enkripsi k*kP untuk mendapatkan titik absis x.
   Menentukan titik awal deskripsi dengan memisahkan header kP dan pesan.
   Hader kP dikalikan dengan k untuk mendapatkan kunci dalam bentuk (x,y).
   Titix x di xor kan dengan pesan terenkripsi untuk mendeskripsikan pesan.
4. Output
   Pesan terdeskripsi/*plaintext*

3.3. Proses Enkripsi

1. *Plaintext* yang akan dienkripsi "STMIKBUDIDARMA"
2. Tentukan kunci publik dengan rumus k*(x,y) dimana k = kunci rahasia dan (x,y) = titik kurva. Titik kurva dapat dipilih secara manual atau dengan rumus persamaan misalnya y^2 = x^3+x+1.
   k = 4 x,y = (8,10)
   public key = (32,40)
3. Tentukan titik kkP untuk mengetahui absis x sebagai awal enkripsi
   k*kP = 4*(32,40)
   Titik kkP : (128,160)
   Selanjutnya titik absisx dari kkP untuk di-xor-kan ke pesan.
   128 -> 10000000

Tabel 1 Konversi pesan ke integer sesuai format ASCII, kemudian jadikan biner. Selanjutnya di xor dengan absis titik kdP.

Plaintext	Ascii	Biner	Xor	Biner	Ascii	*Chipertext*
S	83	01010011	xor	10000000	211	Ó
T	84	01010100	xor	10000000	212	Ô
M	77	01001101	xor	10000000	205	Í
I	73	01001001	xor	10000000	201	É
K	75	01001011	xor	10000000	203	Ë

4. Output pesan terenkripsi dikirim dengan format (header,pesan) = (£# #, ÓÔÍÉË). *Header* adalah *public key*.

3.4. Proses Dekripsi

1. *Chipertext* yang akan didapatkan (£# #, ÓÔÍÉËÂ)
2. pisahkan *header* dan *chipertext*
   Header = £#d# diubah menjadi karakter adalaha 32#40# sebagai titik x,y
   k = 4
3. Tentukan titik kkP dengan rumus k**header* untuk mengetahui absis x dari header
   k**header* = 4*(32,40)
   Titik kkP : (128,160)
   Selanjutnya titik absis x dari kkP untuk di-xor-kan ke pesan.
   128 -> 10000000
   Konversi pesan ke integer sesuai format ASCII, kemudian jadikan biner. Selanjutnya di xor dengan absis titik kdP. Setelah di xor, rangkai lagi menjadi pesan baru yang terenkripsi:

Tabel 2 Hasil Xor

Chipertext	Ascii	Biner	Xor	Biner	Ascii	*Plaintext*
Ó	211	11010011	xor	10000000	83	S
Ô	212	11010100	xor	10000000	84	T
Í	205	11001101	xor	10000000	77	M
É	201	11001001	xor	10000000	73	I
Ë	203	11001011	xor	10000000	75	K
Â	194	11000010	xor	10000000	66	B

4. Output pesan terdeskripsi "STMIK"

3.5. Algoritma Pembangkit Kunci

Berikut ini adalah algoritma enkripsi:

```
Procedure algoritma {
        Input   : k = secret key
                  x = titik kurva x
                  y = titik kurva y
        output  : pk = public key
}
Proses  : kp=k(x*y)^2
          k=k(x*y)
```

3.6 Algoritma Enkripsi

Berikut ini adalah algoritma enkripsi:

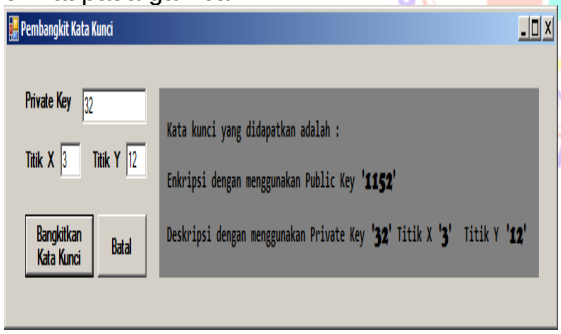
```
Procedure algoritma {
        Input   : t = plaintext
                  k = secret key
                  x = titik kurva x
                  y = titik kurva y
        output  : c = chipertext
}
Proses  : kp=k(x*y)^2
          for (t:=0 to m-n) do
                j = 0
                while (j<n and t[i+j]=t[j]) do
                        ti = t xor (kp);
                end while
                if(t>=n) then
                        c=ti
                end if;
          end for;
```

Menu utama dapat dikatakan sebagai antar muka (*user interface*) antara *user* dan program. Menu utama menampilkan pilihan

menu yang tersedia pada program. Pada menu utama aplikasi enkrip dan depenelitian tersedia 3 pilihan menu yaitu menu enkripsi untuk memenkripsi pesan yang menghubungkan ke *form*enkripsi dan menu depenelitianuntuk mengubungkan ke *form* depenelitian dan menu exit untuk keluar dari program enkripsi dan depenelitian dengan metode *eliptic curve crypthography*. Berikut gambar untuk tampilan menu utama dapat dilihat pada gambar 1.

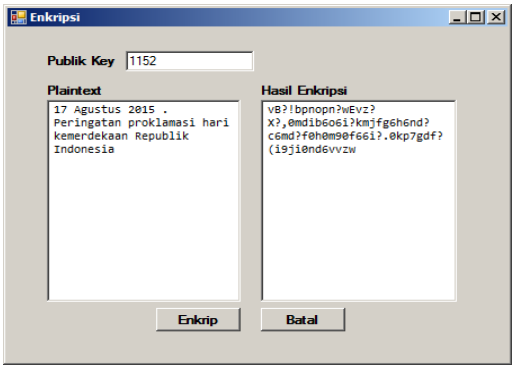
Gambar 1 Menu Utama

Form pembangkit kata kunci digunakan untuk membangkitkan kata kunci *private key* yang akan digunakan untuk mendeskripsikan pesan yang diterima dan *public key* akan digunakan untuk mengenkripsikan pesan yang akan dikirim ke penerima pesan. Berikut gambar untuk tampilan *form*pembangkit kata kunci dapat dilihat pada gambar 2.



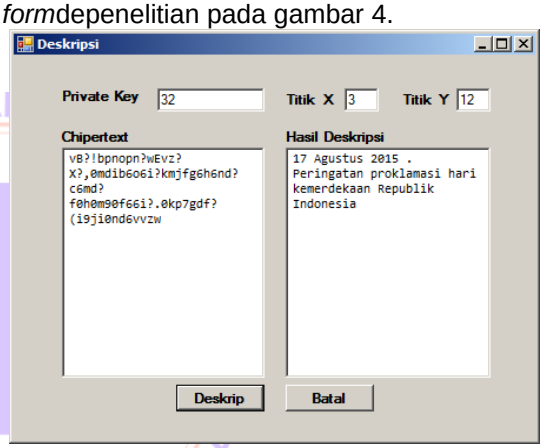
Gambar 2 Form Pembangkit Kata Kunci

Form enkripsi digunakan untuk mengekripsikan text yang akan dikirim ke penerima. *Public key* didapatkan dari *form* pembangkit kunci. *Plaintext* adalah tempat mengetikkan pesan yang akan dienkripsikan. Hasil enkripsi akan tampil ketika tombol enkrip ditekan. Berikut gambar untuk tampilan *form*enkripsi dapat dilihat pada gambar 3.



Gambar 3 Form Enkripsi

Form deskripsi digunakan untuk mendepenelitiankan text yang diterima dari penerima.*Private key* didapatkan dari *form* pembangkit kunci. *Chipertext* adalah tempat mengetikkan pesan yang akan dienkripsikan.Berikut gambar untuk tampilan *form*depenelitian pada gambar 4.



Gambar 4 Form Deskripsi

Tabel 3 Tabel Pengujian Hasil Enkripsi

No.	Plaintext	Kunci enkripsi	Chipertext	Keterangan
1	17 Agustus 2015. Peringatan proklamasi hari kemerdekaan Republik Indonesia	Seharusnya: 1152 Digunakan: 1152	vB?!bpnopn?wEvzX?,0mdlb6o6i?kmjfg6h6nd?c6md?f0h0m90f66i?.0kp7gdf?(i9ji0nd6vvzw	Proses enkripsi pesan berhasil dengan menggunakan kunci yang sesuai
2	Pengibaran bendera Merah Putih oleh anggota PASKIBRA	Seharusnya: 7216 Digunakan: 7216	,0lbd76m6i?70l90m6?|0m6c?,podc?jg0c?6ibbjo6?,!-_(@.!BwvA	Proses enkripsi pesan berhasil dengan menggunakan kunci yang sesuai
3	Aplikasi kriptografi asimetris dengan metode elliptic curve cryptography	Seharusnya: 15552 Digunakan: 15552	!kgdf6nd?fmdkojbm6ad?6ndh0omdn?90ib6i?h0oj90?0ggdkod8?8pmq0?8mtkojbm6kctvzzzw	Proses enkripsi pesan berhasil dengan menggunakan kunci yang sesuai

Tabel 4 Tabel Pengujian Hasil Enkripsi

No.	Chipertext	Kunci deskripsi	Hasil Deskripsi	Keterangan
1	yB?!bpnopn?wEvzX?,0m dlb6o6i?kmjfg6h6nd?c6 md?f0h0m90f66i?.0kp7g df?(i9ji0nd6vvzw	Seharusnya : Private Key : 32 X : 3 Y : 12 Digunakan : Private Key : 32 X : 3 Y : 12	17 Agustus 2015. Peringatan proklamasi hari kemerdekaan Republik Indonesia	Proses deskripsi pesan berhasil dengan menggunakan kunci yang sesuai
2	,0lbd76m6i?70i90m6?]0 m6c?,podc?jg0c?6ibbjo6 ?,!-_(@.!BwvA	Seharusnya : Private Key : 41 X : 8 Y : 22 Digunakan : Private Key : 41 X : 6 Y : 22	-	Proses deskripsi gagal. Karena menggunakan kunci deskripsi yang tidak sesuai. Seharusnya nilai titik X adalah 8
3	!kgdf6nd?fmdkojbm6ad? 6ndh0omdn?90lb6i?h0oj 90?0ggdkod8?8pmq0?8 mtkojbm6kctvzzzw	Seharusnya : Private Key : 27 X : 16 Y : 36 Digunakan : Private Key : 27 X : 16 Y : 36	Aplikasi kriptografi asimetris dengan metode elliptic curve cryptography	Proses deskripsi pesan berhasil dengan menggunakan kunci yang sesuai

KESIMPULAN

Ada beberapa kesimpulan yang dapat diambil penulis setelah merancang dan menyelesaikan penelitian ini adalah :

1. Proses enkripsi dan depenelitian dengan menggunakan metode *Elliptic Curve Cryptography (ECC)*. Dilakukan dengan mengkalkulasikan *private key* dengan titik x dan titix y untuk mendapatkan *public key*. *Private key* digunakan untuk mendepenelitiankan pesan dan *public key* digunakan untuk mengenkripsikan pesan yang akan dikirim kepada penerima.
2. Untuk mengamankan data teks dapat dilakukan dengan enkripsi dan depenelitian dengan metode *Elliptic Curve Cryptography (ECC)*. Metode ini memanfaatkan perbedaan kunci untuk mengnkripsikan dan perbedaan kunci untuk mendepenelitiankan. Sehingga tidak mudah untuk menebak kunci yang digunakan seandainya pesan diterima oleh pihak lain.
3. Cara merancang aplikasi kriptografi ini yaitu dengan menggunakan *Visual Basic* .NET 2008. Memanfaatkan *tools* seperti *textbox, button, form, modul*.

DAFTAR PUSTAKA

[1]. Syah Suhatman Surya, “Kamus Komputer”. Penerbit Rineka Cipta, Surabaya 2002.
[2]. Kadir Abdul, “Pengenalan Teknologi informasi”. Penerbit Andi, Jakarta 2006,
[3]. Arius Doni, “Pengantar Ilmu Kriptografi”. Penerbit Andi Publisher,Jakarta 2006.
[4]. Sugiarti Y, “Analisis Dan Perancangan Unified Modeling Languange” , Penerbit Graha Ilmu, Jakarta 2014.
[5]. Darmayuda Ketut, “Pemrograman Aplikasi Database Dengan Microsoft VisualBasic. Net 2008” ,Penerbit Elex Media Komputindo.
[6]. http///id.Wikipedia org/wiki/ASCI (tanggal akses 07 juni 2015).
[7]. Eko Mardianto, Enkripsi Menggunakan ECC, Politehnik Elektronika Negeri Surabaya.
[8]. http://www.elangsakti.com/2013/07/perhitungan-algoritma-kriptografi-kurva-eliptik-elliptic-curve-cryptography-ecc.html.
[9]. Purwadi, Hendra Jaya, Ahmad Calam, Program Studi Komputer, STMIK Triguna Dharma.
[10]. Lamhot Sitorus, Algoritma dan Pemrograman, Andi Offset, Yogyakarta. 2015