PERANGKAT LUNAK ENKRIPSI Identity-Based Encryption (IBE)

Yarniwati Halawa[1],Swingly Purba[2]

Mahasiswa Prodi Teknik Informatika Fakultas Teknologi Industri, Institut Sains Teknologi TD.Pardede
Dosen Prodi Teknik Informatika Fakultas Teknologi Industri, Institut Sains Teknologi TD.Pardede
Jl.DR. TD.Pardede.No.8 Medan 20153

Email : [1]halawayarni25@gmail.com,[2]swinglypurba@istp.ac.id.

ABSTRAKSI

Identity-Based Encryption (IBE) adalah teknik enkripsi pengamanan data dengan menggunakan kunci asimetris yang mempunyai keistimewaan, yaitu *public-key* yang digunakan dapat berupa sembarang *string* sehingga data dapat diamankan tanpa perlu menggunakan sertifikat. *Public-key* pada *Identity-Based Encryption* ini dapat berupa alamat email, nomor telepon, ataupun suatu kalimat yang menunjukkan identitas dari penerima. Perangkat lunak ini menjelaskan proses dari masing-masing tahapan yang ada dalam metode kriptografi IBE ini. Perangkat lunak ini juga dilengkapi dengan algoritma-algoritma pendukung yang akan digunakan dalam proses perhitungan tahap-tahap kriptografi IBE.Perangkat lunak pembelajaran ini, diharapkan akan menjadi referensi dan studi awal bagi pembaca khususnya mahasiswa yang mempelajari mata kuliah kriptografi untuk mempelajari lebih lanjut mengenai kriptografi IBE.

Kata Kunci : Identity Based Encryption, Kunci Publik, Kunci Privat, Private Key Generator.

1. PENDAHULUAN

Pada dasarnya, alasan digunakannya enkripsi adalah untuk mengamankan data sehingga hanya orang tertentu (misalnya, bob@yahoo.com) atau satu mesin (misalnya, www.yahoo.com) saja yang dapat mengakses data tersebut. Sampai sekarang, teknik enkripsi masih mengandalkan kunci acak yang panjang, yang harus dipetakan untuk identitas tertentu dengan menggunakan dokumen yang ditandatangai secara digital (*digitally signed documents*) yang disebut sertifikat. Masalah sertifikat yang harus selalu diminta *user*, untuk memperoleh kunci private yang baru menjadi masalah dalam teknik enkripsi. *Identity-Based Encryption (IBE)* memberikan pendekatan baru dalam mengatasi masalah pada teknik enkripsi. Dalam IBE, dapat digunakan *string* sembarang sebagai kunci public sehingga data dapat diamankan tanpa perlu menggunakan sertifikat. Pengamanan dilakukan oleh *key server* yang mengendalikan pemetaan identitas ke kunci dekripsi. Rancangan sistem IBE telah menjadi masalah terbuka pada dunia kriptografi. IBE atau *Identity-Based Encryption* merupakan teknik enkripsi dengan menggunakan kunci asimetris yang mempunyai keistimewaan, yaitu *public-key* yang digunakan dapat berupa sembarang *string*. Biasanya, enkripsi menggunakan *public-key* yang rumit dan sulit diingat. *Identity-Based Encryption* menggunakan kunci yang lebih *"user-friendly"*. *Public-key* pada *Identity-Based Encryption* ini dapat berupa alamat email, nomor telepon, ataupun suatu kalimat. Kelebihan lain dari teknik enkripsi ini yaitu tidak diperlukannya penentuan pasangan kunci sebelum melakukan enkripsi. Dengan menggunakan *Identity-Based Encryption*, seseorang dapat mengirimkan email yang telah dienkripsi dengan *public-key* walaupun penerima belum mempunyai bahkan belum pernah mendengar *private-key* sekalipun. Pada saat penerima menerima email yang terenkripsi tersebut, penerima akan menghubungi *Private Key Generator*, yang akan melakukan autentikasi dan memberikan *private-key* untuk membaca email tersebut.

Tujuan dan Manfaat

Tujuan dari perancangan sistem ini adalah menghasilkan perangkat lunak untuk memudahkan

pembelajaran kriptosistem IBE (Identity-BasedEncryption).
Manfaat dari perancangan sistem ini yakni :
1. Memberikan gambaran cara kerja dari IBE.
2. Perangkat Lunak yang dihasilkan dapat digunakan sebagai alat bantu pembelajaran dalam proses belajar mengajar Kriptografi.

LANDASAN TEORI

2.1 Pengenalan Kriptografi

2.1.1 Sejarah Kriptografi

Sejarah Kriptografi pertama sekali dengan menggunakan metode pertukaran posisi untuk mengenkripsi suatu pesan. Dengan sejarah perkembangannya, Julius Caesar dapat mengirimkan pesan yang dibawa oleh hulubalangnya, sengaja mengacak pesan tersebut sebelum diberikan kepada kurir. Hal ini dilakukan untuk menjaga kerahasiaan pesan baik bagi kurir maupun bagi musuh jika kurir tertangkap di tengah perjalanan olehmusuh. Ada seseorang mengatakan bahwa yang dilakukan oleh Julius Caesar dianggap sebagai awal dari kriptografi.

2.1.2 Definisi Kriptografi

Kriptografi berasal dari bahasa Yunani yakni *kriptos* yang artinya tersembunyi dan *graphia* artinya sesuatu yang tertulis, sehingga kriptografi dapat disebut sesuatu yang tertulis secara rahasia.

Kriptografi merupakan bidang ilmu yang didalamnya mempelajari tentang bagai-mana merahasiakan suatu informasi penting ke dalam suatu bentuk yang tidak bisa dibaca oleh siapapun serta mengembalikannya menjadi informasi semula dengan menggunakan berbagai macam teknik sehingga informasi tersebut tidak diketahui oleh pihak manapun yang bukan pemilik atau yang tidak berkepentingan. Sisi lain dari kriptografi ialah kriptanalisis (*Cryptanalysis*) yang merupakan studi tentang bagaimana memecahkan mekanisme kriptografi.

2.1.3 Tujuan Kriptografi

Kriptografi sesungguhnya merupakan studi terhadap teknik matematika yang terkait dengan 4 aspek keamanan dari suatu informasi yaitu kerahasiaan (*confidentiality*), integritas data (*data integrity*), otentifikasi (*authentication*), dan ketiadaan penyangkalan (*non-repudiation*). Keempat aspek tersebut merupakan tujuan utama dari suatu sistem kriptografi yang dapat dijelaskan sebagai berikut :

1. Kerahasiaan (confidentiality)
   Kerahasiaan bertujuan untuk melindungi suatu informasi dari semua pihak yang tidak berhak atas informasi tersebut. Terdapat beberapa cara yang dapat digunakan untuk menjaga kerahasiaan suatu informasi, mulai dari penjagaan secara fisik, misalnya menyimpan data pada suatu tempat khusus, sampai dengan penggunaan algoritma matematika untuk mengubah bentuk informasi menjadi tidak terbaca.
2. Integritas data (*data integrity*)
   Integritas data bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berhak atas informasi tersebut. Untuk menjamin integritas data ini kita harus mempunyai kemampuan untuk mendeteksi terjadinya suatu manipulasi data oleh pihak-pihak yang tidak berkepentingan. Manipulasi data yang dimaksud dalam hal ini meliputi penyisipan, penghapusan, maupun penggantian data.
3. Otentifikasi (*authentication*)
   Otentifikasi merupakan identifikasi yang dilakukan oleh masing-masing pihak yang saling berkomunikasi harus mengindetifikasi satu sama lainnya. Informasi yang didapat oleh suatu pihak dari pihak lain yang harus diidentifikasi untuk memastikan keaslian dari informasi yang diterima. Identifikasi terhadap suatu informasi dapat berupa tanggal pembuatan informasi, isi informasi, waktu kirim, dan hal-hal lainnya yang berhubungan dengan informasi tersebut.
4. Ketiadaan penyangkalan (*non-repudiation*)
   Non-repudiation berfungsi untuk mencegah terjadinya penyangkalan terhadap suatu aksi yang telah dilakukan oleh pelaku aksi itu sendiri. Jika terjadi penyangkalan maka diperlukan suatu prosedur yang melibatkan pihak ketiga untuk menyelesaikan masalah tersebut.

2.2 Algoritma Kriptografi

Algoritma kriptografi adalah fungsi matematika yang digunakan untuk proses enkripsi dan dekripsi. Untuk mengenkripsi sebuah pesan (*plaintext*), maka diterapkan algoritma enkripsi ke pesan tersebut. Untuk mendekripsi sebuah pesan (*ciphertext*), maka diterapkan algoritma dekripsi ke pesan tersebut.

Algoritma kriptografi dapat dibagi menjadi 2 bagian besar, yaitu:

1. Algoritma Kriptografi Klasik, yang terdiri dari *Cipher* Substitusi dan *Cipher* Transposisi.
2. Algoritma Kriptografi Modern, yang terdiri dari Kriptografi Simetris dan Kriptografi Asimetris.

2.6.2 Kriptografi Klasik

Sebelum adanya komputer, kriptografi terdiri dari kriptosistem yang berdasarkan karakter. Berbagai macam algoritma kriptografi melakukan substitusi karakter atau transposisi karakter antara satu dengan yang lainnya. Kriptografi klasik dibagi

menjadi dua bagian yaitu, *Cipher* Subsitusi dan *Cipher* Transposisi.

2.2.1.1 Cipher Substitusi

Cipher subtitusi adalah suatu teknik yang dimana setiap karakter dalam *plaintext* digantikan dengan karakter lain untuk *ciphertext*.

Pada kriptografi klasik ada beberapa dasar dari *cipher* subtitusi yaitu,

1. *SimpleSubstitution Cipher* merupakan salah satu teknik dimana setiap karakter dalam *plaintext* digantikan dengan karakter yang koresponden ke dalam *ciphertext*.
2. *Homophonic Substitution Cipher*, memiliki kesamaan dengan *Simple Substitution Cipher* tetapi sebuah karakter dalam *plaintext* digantikan dengan satu atau beberapa karakter ke dalam *ciphertext*.
3. *A Polyalphabetic Substitution Cipher*dimanaterdiri dari beberapa *Simple Substitution Cipher*.
4. *A Polygram Substitution Cipher* adalah suatu teknik dasar dimana blok karakter dienkripsi dalam kelompok. Contohnya, "ABA" dapat berkoresponden ke "RTQ", "ABB" dan juga berkoresponden ke "SLL", dan lain-lain.

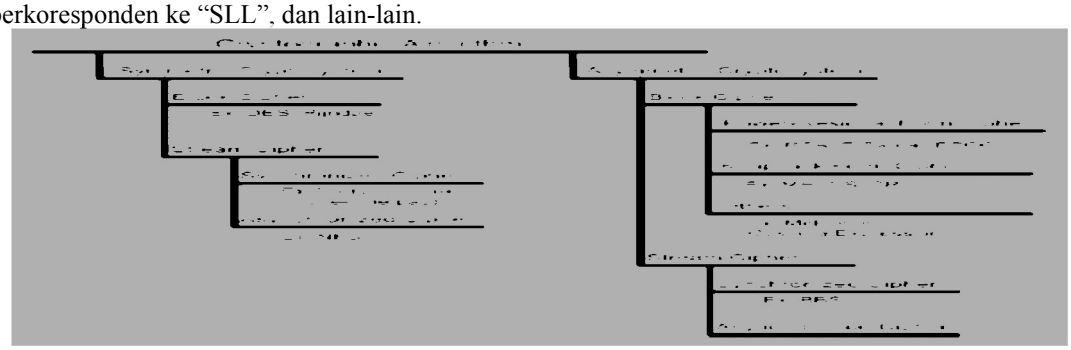
Salah satu *cipher* yang paling tua adalah *cipher Caesar*, yang dikaitkan dengan *Julius Caesar*. Di dalam algoritma ini 'a' menjadi 'D', 'c' menjadi 'F', …. , dan 'z' menjadi 'C'.

2.2.1.2 Cipher Transposisi

Cipher transposisi adalah teknik yang dimana karakter-karakter di dalam *plaintext* tidak berubah, tetapi urutannya diacak. *Cipher* transposisi umum_nya yaitu transposisi kolom, dimana sebuah *plaintext* ditulis secara horizontal pada sebuah kertas grafik dengan lebar yang tetap dan *ciphertext* dibentuk dengan membaca karakter secara vertical.

2.2.2 Kriptografi Modern

Untuk kondisi yang lebih aman, semua algortima kriptografi modern menggunakan kunci atau *key*. Kunci ini dapat berisi banyak sekali peluang nilai. Batas nilai yang mungkin untuk sebuah kunci disebut *keyspace*. Dalam kriptografi yang berbasiskan kunci, terdapat dua buah bentuk umum algoritma, yaitu Algoritma Simetris (*Symmetric Algorithm)* dan Algoritma Asimetris (*Asymmetric Algorithm)*.

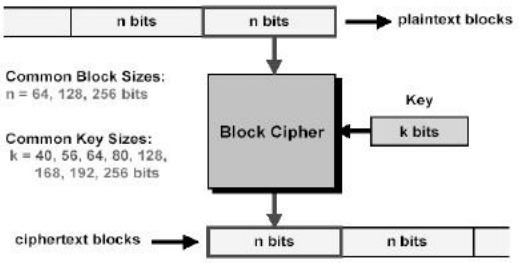


Gambar 2.2 Pengelompokkan Teknik Kriptografi Modern

Algoritma Kriptografi Simetris

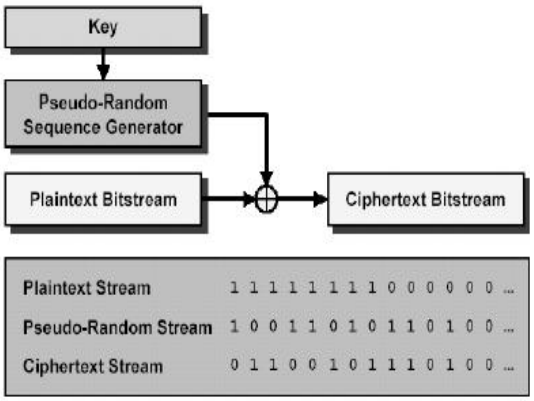
2.2.2.1.1 Block Cipher

Block cipher adalah bentuk algoritma kriptografi simetris yang mentransformasikan satu blok data tersebut dari *plaintext (unencrypted text)* ke dalam satu blok data *ciphertext (encrypted text)* dengan panjang yang sama.



2.2.2.1.2 Stream Cipher

Stream cipher adalah jenis algoritma kriptografi simetris yang dapat dibuat sangat cepat, jauh lebih cepat dibandingkan dengan algoritma *block cipher* yang manapun. Secara umum algoritma *block cipher* digunakan untuk unit *plaintext* yang besar sementara *stream cipher*untuk blok data yang lebih kecil, biasanya ukuran bit. Proses enkripsi terhadap *plaintext* tertentu dengan algoritma *block cipher*menghasilkan *ciphertext* yang sama apabila kunci yang digunakan sama. Dengan *stream cipher*, maka transformasi dari unit *plaintext* yang lebih kecil ini jauh beda antara satu dengan lainnya, tergantung pada kapan unit tersebut ditemukan selama proses enkripsi.



Gambar 2.5 Ilustrasi Algoritma *Stream Cipher*

2.2.2.2 Algoritma Kriptografi Asimetris

Beberapa istilah dalam algoritma kunci publik, yaitu :

1. Kunci publik, yaitu kunci yang diberikan atau disebarkan kepada publik sehingga semua orang akan tahu.
2. Kunci privat, yaitu kunci yang tetap disimpan oleh pemilik kunci.

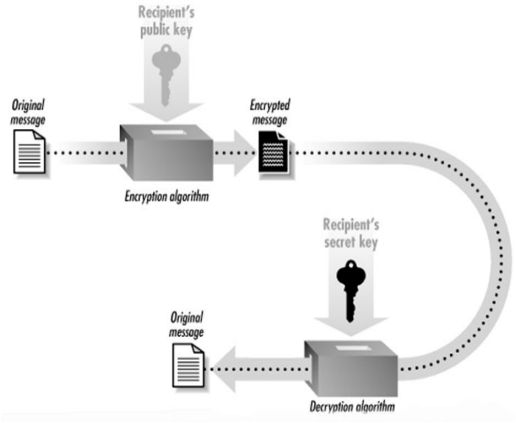
Dalam notasi matematika, proses algoritma kunci publik digambarkan sebagai berikut,

$E_{k1}(P) = C$

$D_{k2}(C) = P$

E_{k1}→ fungsi enkripsi dengan kunci publik

D_{k2}→ fungsi dekripsi dengan kunci privat



Gambar 2.6 Ilustrasi Algoritma Kriptografi Asimetris

2.2.2.3 Perbandingan Kriptografi Simetris dan Asimetris

Baik kriptografi simetris maupun kriptografi asimetris, keduanya mempunyai kelebihan dan kelemahan.

Kelebihan kriptografi simetris adalah :

a. Algoritma kriptografi simetris dirancang sehingga proses enkripsi / dekripsi membutuhkan waktu yang singkat.
b. Ukuran kunci simetris sangat relatif pendek. Algoritma kriptografi simetris dapat digunakan untuk membangkitkan bilangan acak.
c. Algoritma kriptografi simetris dapat disusun untuk menghasilkan *cipher* yang lebih kuat.
d. Otentikasi pengirim pesan langsung diketahui dari *ciphertext* yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja.

Sedangkan, kelemahan dari kriptografi simetris adalah :

a. Kunci simetris harus dikirim melalui saluran yang aman. Maka kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci ini.
b. Kunci harus sering diubah, pada saat sesi komunikasi.

Kelebihan dari kriptografi asimetris adalah :

a. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi namun otentikasi kunci publik tetap terjamin.
b. Pasangan kunci publik ataupun kunci privat tidak perlu diubah, bahkan dalam periode waktu yang panjang.

c. Digunakan untuk mengamankan pengiriman kunci simetris.
d. Beberapa algoritma kunci publik dapat digunakan untuk memberikan tanda tangan digital pada pesan.

Sedangkan, kelemahan dari kriptografi asimetris adalah :

a. Enkripsi dan dekripsi data umumnya lebih lambat daripada sistem simetris,di karenakan enkripsi dan dekripsi menggunakan bilangan yang besar dan melibatkan operasi perpangkatan yang besar.
b. *ciphertext* lebih besar ukuran-nya daripada*plaintext* (bisa dua sampai empat kali ukuran *plaintext*).
c. Ukuran kunci relatif lebih besar di banding ukuran kunci simetris.
d. Karena kunci publik sudah dikenal secara luas dan juga digunakan oleh setiap orang, maka *ciphertext* tidak memberikan informasi mengenai otentikasi pengirim.
e. Tidak ada algoritma kunci publik yang sudah terbukti aman (sama seperti *block cipher*). Kebanyakan algoritma mendasarkan keamanannya pada sulitnya memecahkan persoalan-persoalan asimetrik (seperti pemfaktoran, logaritmik, dan sebagainya) yang menjadi dasar pembangkitan kunci. Kriptografi kunci publik juga mengalami ketidak nyamanan dari serangan *man-in-the-middle attack*. Orang di ”tengah” melakukan intersepsi komunikasi lalu berpura-pura sebagai salah satu pihak yang berkomunikasi untuk mengetahui informasi rahasia.

2.3 Landasan Matematis Kriptografi

2.3.1 Greatest Common Divisor (GCD)

Salah satu metode yang dapat digunakan untuk menghitung GCD dari dua buah bilangan adalah algoritma *Euclidean* yang ditulis oleh Euclid dalam bukunya ‘*Elements*’ sekitar 300 tahun sebelum masehi. Knuth mendeskripsikan algoritma ini dengan beberapa modifikasi modern seperti berikut :

1. Set P sebagai bilangan pertama A dan Q sebagai bilangan kedua B.
2. Lakukan proses berikut hingga Q = 0.
   a. Set R = P Mod Q.
   b. Set P = Q.
   c. Set Q = R.
3. GCD(A, B) = P.

2.3.2 Bilangan Prima

Bilangan prima merupakan integer yang lebih besar dari 1 yang hanya mempunyai faktor 1 dan dirinya sendiri. Selain itu tidak ada bilangan lain yang habis dibagi dengan bilangan tersebut. Jadi 2 merupakan bilangan prima begitu juga dengan 73, 2521, 2365347734339, dan $2^{756839} - 1$. Terdapat jumlah tak terbatas dari bilangan prima. Kriptografi, khususnya kriptografi kunci publik, menggunakan bilangan prima yang besar (512 bit dan bahkan lebih besar lagi)

2.3.3Grup

Grup adalah sebuah obyek matematika abstrak yang terdiri dari sebuah himpunan G dan sebuah operasi biner $*$ yaitu :

1. $a * (b * c) = (a * b) * c$ untuk $a,b,c \in G$ (asosiatif).
2. terdapat sebuah elemen $e \in G$ sehingga $e*a = a*e =$ untuk setiap $a \in G$
3. untuk setiap $a \in G$ terdapatsebuah elemen $b \in G$ sehingga $b*a = a*b = e$
   b disebut dengan invers a

2.3.4 Randomize

Randomize atau bilangan acak berarti suatu bilangan yang diambil dari sekumpulan bilangan dimana tiap-tiap elemen dari sekumpulan bilangan ini mempunyai peluang yang sama untuk terambil. Contoh kumpulan dari bilangan berjumlah n, maka masing-masing elemen mempunyai peluang 1/n untuk terambil. Jika jumlah bilangan yang akan diambil lebih dari satu, maka masing-masing proses pengambilannya haru bersifat bebeas secara statistic. Proses pengambilan bilangan acak tersebut merupakan suatu random event (peristiwa acak), yang berarti suatu peristiwa dimana proses dan hasilnya tidak dapat diprediksi.

2.4 Kriptosistem Kurva Elips (*Elliptic Curves Cryptosystem*)

Pada tahun 1985, Neil Koblitz dan Viktor Miller memproposalkan secara terpisah kriptosistem kurva elips (*Elliptic Curves Cryptosystem - ECC*) dengan menggunakan masalah logaritma diskrit pada setiap titik kurva elips yang disebut dengan ECDLP (*Elliptic Curves Discrete Logarithm Problem*). Kriptosistem kurva ellips dapat digunakan dalam beberapa keperluan seperti :

- Skeme enkripsi (ElGamal ECC)
- Untuk Tanda tangan digital (ECDSA – *Elliptic Curves Digital Signature*)
- Pada Protokol pertukaran kunci (Diffie Hellman ECC)

Untuk saat ini ada tiga macam sistem kriptografi kunci publik yang aman dan efisien dikelompokanberdasarkan suatu permasalahan matematis, yaitu :

- **Sistem Pemfaktoran Bilangan Integer (*Integer Factorization Systems*)**
  Tipe ini menngunakan masalah matematis yang disebut *Integer Factorization Problem* (IFP). Apabila diberikan bilangan integer n yang merupakan hasil kali dua buah bilangan prima, maka yang harus dicari yaitu kedua bilangan prima p dan q yang merupakan faktor n, sehingga $n = p * q$.Dengan cara ini

tentunya akan menyebabkan kesulitan menghitung faktor integer yang besar.

- **Sistem Logaritma Diskrit (*Discrete Logarithm Systems*)**
  Jika dipilih bilangan prima *p* dan diberikan bilangan integer *g* antara 0 dan *p*-1 serta *y*maka merupakan bilangan pemangkatan dari *g* sehingga :

$y = g^x \pmod p$

untuk beberapa *x*. Masalah logaritma diskrit pada modulo *p* adalah untuk mencari bilangan *x* jika diberikan pasangan bilangan *g* dan *y*. Dengan ini menyebabkan kesulitan dalam menghitung $x = (\log b) \bmod p$.

- **Kriptosistem Kurva Elips (*Elliptic Curves Cryptosystem*)**
  Pada sistem ini dapat digunakan dalam masalah logaritma diskrit kurva elips dengan menggunakan grup kurva elips. Struktur kurva elips digunakan untuk grup operasi matematis dalam melangsungkan proses enkripsi dan deskripsi. Hal ini menyebabkan kesulitan dalam menghitung *k* jika diketahui *Q* dan *P* dimana *Q* = *k P*.

2.5 SHA (Secure Hash Algorithm)

NIST bersama dengan NSA mendesain *Secure Hash Algorithm* (SHA) untuk digunakan sebagai komponen *Digital Signature Standard* (DSS). Standard hash adalah *Secure Hash Standar* (SHS) dengan SHA sebagai algoritma yang digunakan. Jadi, SHS adalah standar sedangkan SHA adalah algoritma.

2.5.1Algoritma SHA-1

Berikut adalah algoritma SHA-1 :

a. Message Digest dihitung dengan menggunakan pesan yang di-padded terakhir. Penghitungan menggunakan dua buffer masing-masing buffer terdiri dari lima 32 bit kata dan sequence 80 buah kata masing-masing 32 bit. Lima buffer pertama diberi nama A, B, C, D, E sedangkan lima buffer kedua diberi nama H_0, H_1, H_2, H_3, dan H_4. Kemudian pada 80 kata yang berurutan diberi nama W_0, W_1, …, W_{79} dan dalam penghitungan juga memakai buffer TEMP.

b. Untuk menghasilkan message digest, mula-mula Mi dibagi ke dalam 16 blok @ 32 bit : $M^{(1)}$, $M^{(2)}$, …, $M^{(15)}$.Caranya : 32 bit pertama dari blok pesan dimasukkan ke $M_0^{(i)}$, lalu 32 bit berikutnya ke $M_1^{(i)}$ dan selanjutnya hingga $M_{15}^{(i)}$. Proses dilakukan dalam 80 tahap.

c. Mengisialisasi Nilai Hash (dalam bentuk hex) :

H_0 = 67452301
H_3 = 10325476
H_1 = EFCDAB89
H_4 = C3D2E1F0
H_2 = 98BADCFE

d. Lakukan proses menghitung M_1, M_2, …, M_n dengan cara membagi M_i ke dalam 16 kata W_0, W_1, …, W_{15} dimana W_0 merupakan left most.

e. Hitung nilai : For t = 16 to 79

$$W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$$

f. Inisialisasi 5 variabel A, B, C, D, dan E dengan nilai Hash :
$A = H_0$; $B = H_1$; $C = H_2$; $D = H_3$; $E = H_4$.

g. Hitung nilai : For t = 0 to 79

$$TEMP = S^5(A) + f_t(B,C,D) + E + W_t + K_t$$

$E = D$; $D = C$; $C = S^{30}(B)$; $B = A$; $A = TEMP$.

h. Hitung Nilai dari Hash :

$H_0 = H_0 + A$; $H_1 = H_1 + B$; $H_2 = H_2 + C$; $H_3 = H_3 + D$; $H_4 = H_4 + E$.

Hasil dari message digest adalah 160 bit dari pesan, M adalah :

$H_0\ H_1\ H_2\ H_3\ H_4$

2.6 Metode IBE (Identity-Based Encryption)

2.6.1 Sejarah IBE

Konsep IBE ditemukan pada tahun 1984 oleh Adi Shamir dalam rangka mengatasi masalah autentikasi kunci public. Idenya adalah untuk menghindari kebutuhan autentikasi dengan cara kunci publik yang digunakan berhubungan langsung dengan identitas *user*. Kunci publik *user*dapat dihasilkan langsung dari informasi publik sudah tersedia untuk mengidentifikasikan user tersebut secara unik. Informasi ini merupakan identitas digital *user*. Bergantung pada aplikasi, identitas ini dapat berupa (kombinasi dari) nama *user*, nomor kartu identitas, nomor telepon, alamat email, atau informasi yang mungkin lainnya. Dengan demikian, kunci publik *user* telah siap tersedia untuk siapapun yang dapat mengetahui identitasnya sehingga tidak diperlukan lagi pencarian kunci public sehingga menghilangkan kebutuhan akan sertifikat seperti pada PKI. Bagaimanapun, realisasi hubungan antara *user* dengan identitas digitalnya cukup sulit.

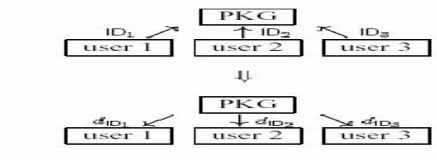
2.6.3 Algoritma IBE

Secara umum, teknik IBE terdiri dari empat algoritma, yaitu :

- *Setup*, yaitu mengambil parameter kemanan sebagian input untuk menentukan parameter sistem dan *master key*. Parameter sistem meliputi

deskripsi ruang *plaintext* dan ruang *ciphertext.* Parameter sistem akan dipublikasikan, sedangkan *master key* hanya boleh diketahui oleh PKG. Algoritma ini dijalankan oleh PKG.

- *Extract*, merupakan proses pembuatan kunci privat d_{ID} yang bersesuaian dengan identitas kunci publik (*string* yang digunakan) dari parameter sistem, *master key*, dan identitas (*string* sembarang) ID є {0,1}. Algoritma ini juga dijalankan oleh PKG di saat *user* meminta kunci privatnya (dengan memberikan *string* yang digunakan untuk menghasilkan kunci publik) untuk mendekripsi pesan. Dalam hal ini, *user* harus membuktikan kepada PKG bahwa dirinya benar-benar pemilik dari identitas atau *string* tersebut, seperti yang ditunjukkan pada gambar 2.12.
- *Encrypt*, yaitu mengenkripsi pesan *M* untuk *user* yang dituju menggunakan kunci publik dan parameter sistem menghasilkan *ciphertext.* Algoritma ini hanya dapat dijalankan oleh *user.*
- *Decrypt*, yaitu mendeskripsi *ciphertext C* dengan menggunakan kunci privat d_{ID} dan parameter sistem menghasilkan *plaintext.* Seperti algoritma *encrypt*, yang dijalankan oleh *user.*



Gambar 2.7 Permintaan kunci privat

2.6.4 Teknik *Identity-Based Encryption* Boneh-Franklin

Teknik IBE oleh Boneh dan Franklin merupakan teknik IBE pertama yang efisien dan aman. Untuk mengenkripsi pesan, pengirim menggunakan bilinear map dengan tujuan menggabungkan identitas penerima, parameter sistem dari PKG dan *master key* menjadi kunci untuk enkripsi. Penerima pesan menghasilkan kunci untuk dekripsi dengan menggunakan bilinear map untuk menggabungkan kunci privat penerima dan parameter publik yang dikirimkan bersama *ciphertext*.

Secara umum, teknik ini dapat dideskripsikan sebagai berikut:

1. *Setup*
   Untuk menginisialisasi IBE, PKG mengambil 2 titik pada kurva elips, yaitu: sebuah *master key s*, dan sebuah titik *P* menggunakan generator angka acak. Kemudian,parameter publik, yaitu *P* dan $s \bullet P$ didistribusikan ke semua *user*, biasanya melaluisertifikat server. *Master key* **s** juga dapat di-*share* secara rahasia sehingga tidak ada serveryang dapat berkompromi dengan sistem.
2. *Extract*
   Pada saat Bob menerima pesan Alice, Bob belum mempunyai kunci *k* untuk mendekripsi. Untuk memperoleh kunci tersebut, Bob melakukan autentikasi ke PKG. Setelah Bob diautentikasi, server menghitung $s \bullet ID_{Bob}$danmemberikannya kepada Bob. Nilai ini merupakan kunci privat Bob.
3. *Encrypt*
   Untuk mengirim pesan ke Bob, Alice memetakan identitas Bob (misalnya bob@yahoo.com) ke suatu titik pada kurva elips ID_{Bob}. Kemudian Alice memilih suatu angka acak r dan menghitung kunci *k*, sebagai berikut:
   $k = Pair (r \bullet ID_{Bob} , s \bullet P)$
   Setelah mendapatkan kunci, Alice mengirimkan pesan yang dienkripsi dengan *k*, E_k[Pesan], kepada Bob. Selain itu dikirimkan juga hasil perkalian $r \bullet P$.
4. *Decrypt*
   Setelah menerima pesan beserta hasil perkalian $r \bullet P$, Bob dapat memperoleh kunci *k* dengan menghitung:
   $k = Pair (s \bullet ID_{Bob} , r \bullet P)$
   Kunci *k* di atas, oleh karena sifat *bilinear map*, sama dengan kunci yang digunakan Alice untuk mengenkripsi pesan, yaitu:
   $k = Pair (r \bullet ID_{Bob} , s \bullet P)$
   Dengan kunci *k* tersebut, Bob dapat mendekripsi pesan yang diterimanya. Oleh karena hanya Bob yang mengetahui kunci privatnya, yaitu $s \bullet ID_{Bob}$, maka tidak ada orang lain yang dapat menghitung *k*.

PEMBAHASAN DAN PERANCANGAN

3.1 Pembahasan

3.1.1 Persyaratan Perangkat Lunak

Perangkat lunak pembelajaran IBE memiliki persyaratan sebagai berikut :

1. Perangkat lunak dapat menerima *plaintext* (*message*) maksimum 48 karakter dan *kunci* (ID tujuan) dapat di pilih sesuai pilihan :
   a. E-Mail, karakter yang bisa diterima minimal 15 dan berakhiran ".com".
   b. Hp Number, karakter yang bisa diterima minimal 11 dan maksimal 12.
   c. Phone Number, karakter yang bisa diterima minimal 5 dan maksimal 7.
   d. Address, karakter tidak dibatasi.
   e. Id Card Number, karakter yang bisa diterima 16.
2. Perangkat lunak harus dapat menampilkan tahap pembentukan kunci, tahap enkripsi dan tahap dekripsi.
3. Perangkat lunak pada tahap setup mengambil 2 titik pada kurva elips menggunakan metode angka acak dan setiap pengambilan angka acak pada program harus 5 digit.

4. Perangkat lunak akan menyediakan teori-teori dasar mengenai IBE dan teori-teori pendukung SHA-1.
5. Kecepatan visualisasi proses pada perangkat lunak dapat diatur.
6. Perangkat lunak harus dapat menampilkan setiap tahapan proses yang dilakukan oleh program.
7. Perangkat lunak menggunakan nama 'Alice' sebagai *user* pertama atau *user* pengirim pesan, nama 'Bob' sebagai *user* kedua atau *user* penerima pesan pertama, dan nama 'Charlie' sebagai *user* ketiga atau *user* penerima pesan kedua.
8. Perangkat lunak akan menampilkan 2 model pengiriman pesan, yaitu model 2 user dan model 3 user. Model 2 user, Alice akan mengirimkan pesan kepada Bob. Sedangkan model 3 user, Alice akan mengirimkan pesan kepada Bob, kemudian pesan akan diproses kembali dan kemudian dikirim kepada Charlie. Pesan yang diproses kembali, menggunakan proses yang sama seperti yang dilakukan Alice.
9. Perangkat lunak menggunakan keyboard sebagai input pesan atau pesan dimasukkan oleh *user*.
10. Untuk model 3 user, *kunci* (ID tujuan) antara user yang berbeda tidak boleh sama.
11. *User* diasumsikan telah memahami dasar matematika kriptografi seperti operasi XOR, perkalian modulo, penjumlahan modulo, konversi antar basis bilangan dan konversi dari bilangan ke *ASCII Code* yang mencakup biner ke desimal, heksadesimal ke desimal, biner ke heksadesimal, biner ke *ASCII Code*, heksadesimal ke *ASCII Code* dan sebaliknya.

3.1.2 Pembahasan Tahapan IBE

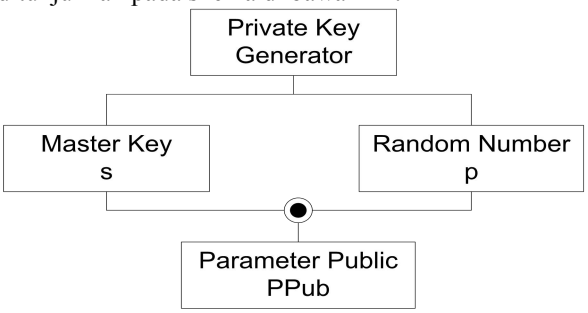
3.1.2.1 Tahap Pembentukan Kunci

Dalam tahap pembentukan kunci, IBE akan membangkitkan sejumlah parameter dan akan melakukan autentikasi dengan identitas penerima. Tahapan-tahapan ini, dibagi menjadi 2, yaitu tahap setup dan tahap extract.

3.1.2.1.1 Tahap Setup

Pada tahap pertama dalam IBE, sistem akan membangkitkan sejumlah parameter yang akan digunakan dalam proses enkripsi dan dekripsi. Tahap setup dalam IBE dijalankan oleh *Private Key Generator* (PKG). PKG akan menghasilkan sebuah bilangan acak p dan juga akan menghasilkan sebuah *Master Key* s. Kedua bilangan ini dihasilkan denganmenggunakan fungsi random, yang menghasilkan bilangan acak. Kemudian kedua nilai akan dikali menghasilkan sebuah bilangan PPub.

Proses untuk menghasilkan nilai acak p dan s ditunjukkan pada skema di bawah ini.



Gambar 3.1 Skema tahap setup IBE

3.1.2.2 Tahap Enkripsi

Dalam tahap enkripsi, sistem akan melakukan proses enkripsi pesan yang akan dikirimkan ke tujuan atau penerima pesan. Pada tahap ini, sistem membutuhkan masukan parameter acak r, identitas tujuan atau penerima (ID Bob), dan pesan yang akan dienkripsi.

3.1.2.3 Tahap Dekripsi

Dalam tahap dekripsi, sistem akan melakukan proses dekripsi pesan yang diterima. Pada tahap ini, sistem membutuhkan masukan identitastujuan (ID Bob), *ciphertext* dan *privatekey* yang didapat dari PKG. Tahap dekripsi mengubah *ciphertext* yang diterima menjadi *plaintext* atau pesan asli.

3.2 Perancangan

Perangkat lunak pembelajaran IBE ini dirancang dengan menggunakan beberapa objek (*tools*) yang terdapat di dalam bahasa pemrograman *Microsoft Visual Basic.Net* seperti:

1. *Command button*, yang digunakan sebagai tombol eksekusi.
2. *Label*, yang digunakan untuk menampilkan keterangan.
3. *Textbox*, yang digunakan sebagai tempat penginputan data *message*/pesan, identitas tujuan, *ciphertext,* dan *plaintext*.
4. *Common dialog control*, yang digunakan untuk menampilkan dialog *save*.
5. *Image*, digunakan untuk menampilkan gambar proses pengiriman data.
6. *Timer*, digunakan untuk mengatur interval waktu dari setiap langkah dalam proses pembelajaran.
7. *Combo box*, digunakan untuk menyediakan pilihan kecepatan animasi.
8. *Listbox*, digunakan sebagai tempat menampilkan tahapan-tahapan proses yang dilakukan sistem.

3.2.2 Perancangan Tampilan

Perangkat lunak pembelajaran IBE ini memiliki beberapa rancangan tampilan seperti:

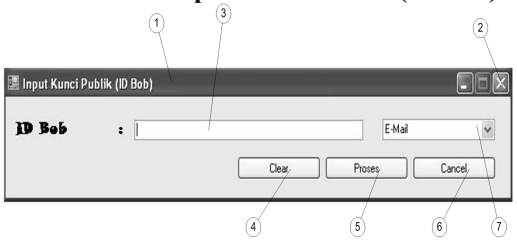
1. *Form* 'Menu'.

2. *Form* 'Input Kunci Publik (ID Bob)'.
3. *Form* 'Input Data Enkripsi (2 user)'.
4. *Form* 'Input Data Dekripsi (2 user)'.
5. *Form* 'Input Kunci Publik (ID Bob & ID Charlie)'.
6. *Form* 'Input Data Enkripsi (3 user – Alice ke Bob)'.
7. *Form* 'Input Data Dekripsi (3 user – Bob)'.
8. *Form* 'Input Data Enkripsi (3 user – Bob ke Charlie)'.
9. *Form* 'Input Data Dekripsi (3 user – Charlie)'.
10. *Form* 'Proses Pembentukan Kunci (2 user)'.
11. *Form* 'Proses Enkripsi (2 user)'
12. *Form* 'Proses Dekripsi (2 user)'
13. *Form* 'Proses Pembentukan Kunci (3 user)'.
14. *Form* 'Proses Enkripsi (3 user – Alice ke Bob)'.
15. *Form* 'Proses Dekripsi (3 user – Bob)'.
16. *Form* 'Proses Enkripsi (3 user – Bob ke Charlie)'.
17. *Form* 'Proses Dekripsi (3 user – Charlie)'.
18. *Form* 'Sejarah'.
19. *Form* 'Teori'.
20. *Form* 'SHA-1'.
21. *Form* 'Team'.

3.2.2.1 Form Menu

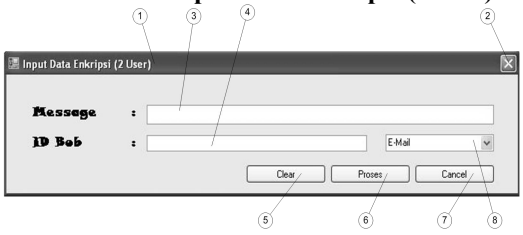
Gambar 3.5 Rancangan Form Menu

3.2.2.2 Form Input Kunci Publik (ID Bob)



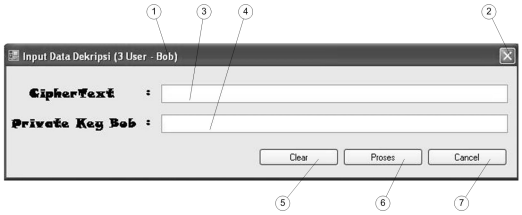
Gambar 3.7 Rancangan Form Input Kunci Publik (ID Bob)

3.2.2.3 Form Input Data Enkripsi (2 user)



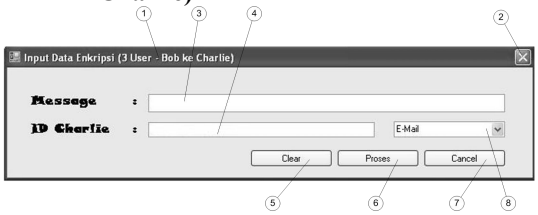
Gambar 3.11 Rancangan Form Input Data Enkripsi (3 user – Alice ke Bob)

3.2.2.4 Form Input Data Dekripsi (3 user - Bob)



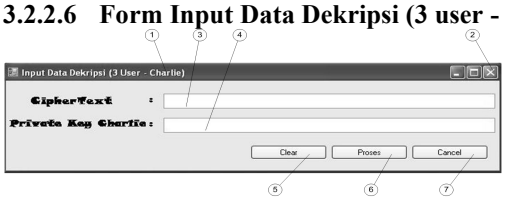
Gambar 3.12 Rancangan Form Input Data Dekripsi (3 user - Bob)

3.2.2.5 Form Input Data Enkripsi (3 user – Bob ke Charlie)



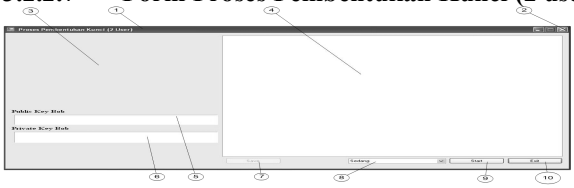
Gambar 3.13 Rancangan Form Input Data Enkripsi (3 user – Bob ke Charlie)

3.2.2.6 Form Input Data Dekripsi (3 user - Charlie)



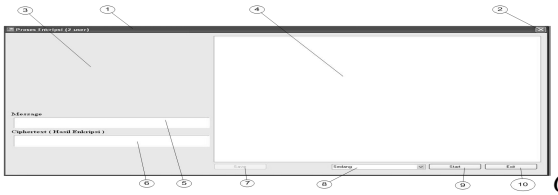
Gambar 3.14 Rancangan Form Input Data Dekripsi (3 user – Charlie)

3.2.2.7 Form Proses Pembentukan Kunci (2 user)



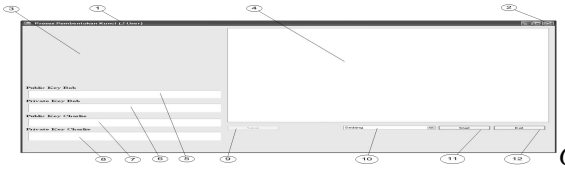
Gambar 3.15 Rancangan Form Proses Pembentukan Kunci (2 user)

3.2.2.8 Form Proses Enkripsi (2 user)



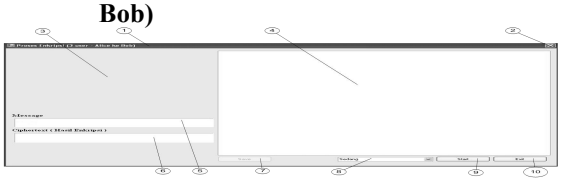
Gambar 3.16 Rancangan Form Proses Enkripsi (2 user)

3.2.2.9 Form Proses Pembentukan Kunci (3 user)



Gambar 3.18 Rancangan Form Proses Pembentukan Kunci (3 user)

3.2.2.10 Form Proses Enkripsi (3 user – Alice ke Bob)



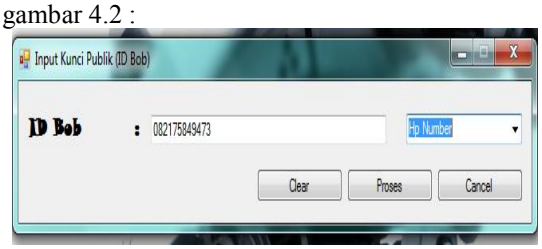
Gambar 3.19 Rancangan Form Proses Dekripsi (3 user - Bob)

IMPLEMENTASI SISTEM

. Untuk memulai perangkat lunak ini, jalankan file ”IBE.EXE”, maka akan tampil *form* ’Menu’, seperti yang ditunjukkan pada gambar 4.1 :

Gambar 4.1 Tampilan Form Menu

Di dalam menu utama, terdapat menu proses, menu about, dan menu exit. Untuk pembelajaran IBE, maka dipilih sub menu proses, kemudian dipilih submenu untuk 2user atau sub menu untuk 3 user. Jika memilih sub menu 2 user, kemudian untuk membuat kunci, maka dipilih sub menu Pembentukan Kunci, maka akan ditampilkan *form* ’Input Kunci Publik (ID Bob)’, seperti pada gambar 4.2 :



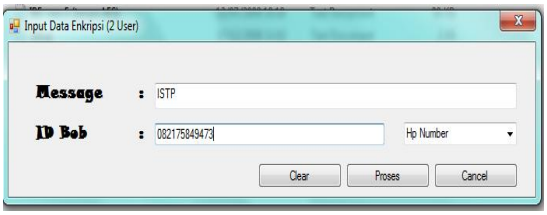
Gambar 4.2 Tampilan Form Input Kunci Publik (ID Bob)

Misalkan :
ID Bob : 082175849473 (Hp Number)

Setelah itu, klik tombol proses untuk memulai proses pembentukan kunci, maka akan tampil *form* ‘Proses Pembentukan Kunci (2 user)’ seperti yang ditampilkan pada gambar 4.3 :

Gambar 4.3 Tampilan Proses Pembentukan Kunci (2 user)

Setelah itu, pilih sub menu Enkripsi, akan keluar *form* ‘Input Data Enkripsi (2 user)’, seperti yang ditampilkan pada gambar 4.4 :



Gambar 4.4 Tampilan Form Input Data Enkripsi (2 user)

Misalkan :
Message : ISTP
ID Bob : 082175849473 (Hp Number)

Setelah itu, klik tombol proses untuk memulai proses enkripsi, maka akan tampil *form* ‘Proses Enkripsi (2 user)’ seperti yang ditampilkan pada gambar 4.5 :

Gambar 4.5 Tampilan Proses Enkripsi (2 user)

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menyelesaikan tugas akhir ini, penulis menarik beberapa kesimpulan sebagai berikut :

1. Perangkat lunak mampu menampilkan proses kerja dari algoritma kriptografi IBE secara terperinci tahapan demi tahapan dan juga fasilitas pengaturan kecepatan perhitungan serta animasi proses, sehingga dapat digunakan untuk membantu pemahaman algoritma kriptografi IBE.
2. Perangkat lunak juga menyediakan fasilitas *save* yang dapat menyimpan proses kerja dan hasil perhitungan dari algoritma kriptografi IBE, sehingga dapat digunakan untuk membantu pemahaman algortima kriptografi IBE.

5.2 Saran

Penulis ingin memberikan beberapa saran yang mungkin berguna untuk pengembangan perangkat lunak lebih lanjut, yaitu :

1. Perangkat lunak dapat dikembangkan lebih lanjut dengan merancang aplikasi enkripsi dan dekripsi *file* dengan menggunakan algoritma kriptografi IBE.
2. Perangkat lunak dapat ditambahkan beberapa animasi gambar 3D dan efek suara agar proses pembelajaran menjadi lebih menarik.

DAFTAR PUSTAKA

1. Kurniawan. J, Kriptografi: Keamanan Internet dan Jaringan Komunikasi, Penerbit Informatika, Bandung.

2. Schneier. B, Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition, Penerbit John Wiley & Sons, Inc, USA.
3. Jennifer Seberpy, Jojef Pieprzyk, ***Cryptography : An Introduction to Computer Security***.
4. Bruce Schneier, ***Applied Crytography***, *Second Edition*, John Willey and Sons Inc..
5. http://crypto.stanford.edu/ibe/ Tanggal Akses : 30 April 2021
6. http://crypto.stanford.edu/~dabo/papers/ccaibejour.pdf Tanggal Akses : 30 April