Implementasi *Identity Based Encryption* Untuk Keamanan Komunikasi *Email*

A. Ais Prayogi

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if11035@students.if.itb.ac.id

Abstrak

Makalah ini membahas studi dan implementasi *Identity Based Cryptography (IBC)* untuk melakukan proses pengamanan data pada *email*. *IBC* merupakan suatu sistem kriptografi kunci publik baru yang memanfaatkan *string* sembarang yang merepresentasikan identitas seseorang atau suatu entitas sebagai kunci publik. Sedangkan kunci privat dibangkitkan oleh *trusted third party* yaitu *Private Key Generator* (*PKG*). Pembangunan perangkat lunak terdiri dari aplikasi *plug-in* pada *Microsoft Outlook* dan aplikasi *web.* Aplikasi *plug-in* yang dikembangkan berfungsi untuk melakukan pengamanan pada *email* dengan enkripsi dan pemberian tanda tangan digital, sedangkan aplikasi *web* berfungsi sebagai *PKG*. Pengamanan data pada *email* dilakukan pada bagian *body* dan *attachment email*. Perangkat lunak dikembangkan dengan menggunakan *tool* pengembangan *Visual Studio .NET 2003* dengan bahasa pemrograman *C#* dan *ASP.NET* dalam lingkungan pengembangan sistem operasi *Windows*. Berdasarkan pengujian yang dilakukan, perangkat lunak yang dibangun mampu melakukan proses enkripsi, dekripsi, pemberian tanda tangan digital dan verifikasi pada *email* berdasarkan skema *IBC*.

***Kata kunci**: Identity Based Cryptography, email, sistem kriptografi kunci publik, kunci publik, , kunci privat, trusted third party, Private Key Generator, plug-in, Microsoft Outlook, enkripsi, dekripsi, tanda tangan digital, body, attachment, verifikasi.*

1. Pendahuluan

Tahun 1984, Shamir mengajukan suatu bentuk sistem kriptografi kunci publik yang baru untuk mengurangi kerumitan yang terdapat dalam struktur kunci publik tradisional. Sistem kriptografi tersebut adalah *Identity Based Cryptography* (IBC). Pada IBC, kunci publik seseorang merupakan string sembarang yang merepresentasikan identitas pemilik kunci publik seseorang. String tersebut berupa nomor kartu identitas, username pada lingkungan Linux, alamat email, nomor telepon, alamat rumah atau informasi lainnya yang dapat mewakili secara unik identitas suatu entitas dalam hal ini adalah pemilik kunci.

Konsep dasar IBC adalah untuk menghindari adanya otentikasi terhadap kunci publik seperti dalam *Public Key Infrastructure* (PKI). Satu-satunya otentikasi yang diperlukan adalah otentikasi seorang user untuk mendapatkan kunci privatnya. Dengan demikian, penggunaan IBC dapat mengurangi secara signifikan kerumitan yang diperoleh jika dibandingkan dengan kunci publik konvensional atau PKI. Dengan menggunakan identitas seseorang sebagai kunci publik akan menghilangkan kebutuhan sertifikat kunci publik yang dikeluarkan oleh *Certificate Authorities* (CA).

Sejak pertama kali diusulkan, telah muncul beberapa ide dan konsep untuk mewujudkan IBC. Namun baru pada tahun 2001, melalui konsep IBC yang diajukan oleh Boneh dan Franklin [2], didapatkan solusi IBC yang benar-benar dapat diimplementasikan secara praktis dan mempunyai tingkat keamanan yang cukup kuat.

Munculnya IBC telah memberikan sebuah solusi baru dalam melakukan komunikasi melalui email dengan aman. Beberapa keuntungan yang diberikan dengan penggunaan IBC dalam komunikasi email adalah [4]:

1. Pengguna email tidak perlu melakukan pertukaran kunci secara eksplisit baik kunci simetris maupun kunci publik.
2. Daftar kunci publik tidak perlu di-maintain secara khusus.
3. Pihak ketiga, dalam hal ini *Private Key Generator* (PKG), hanya diperlukan pada fase awal saja untuk menentukan parameter kunci publik dan pembangkitan kunci privat.

2. Ruang Lingkup

Makalah membahas tentang skema *Identity Based Cryptography* (IBC) serta proses implementasi IBC sebagai *plug-in* dan aplikasi *web* untuk *Microsoft Outlook* .

3. Skema Identity Based Cryptography

Skema *Identity Based Cryptography* (IBC) yang digunakan dalam implementasi terbagi atas 2 algoritma yaitu algoritma enkripsi-dekripsi dan algoritma tanda tangan digital-verifikasi. Untuk algoritma enkripsi-dekripsi, digunakan skema enkripsi-dekripsi *Boneh-Franklin-Identity Based Encryption* (BF-IBE). Skema BF-IBE didasarkan pada dua permasalahan kriptografi yaitu *Bilinear Diffie-Helman Problem (BDHP) dan Elliptic Curve Discrete Logarithmic Problem (ECDLP).* Secara matematis skema dasar BF-IBE dijelaskan sebagai berikut :

1. Setup, pada tahap ini sistem akan membangkitkan sejumlah parameter yang akan digunakan dalam proses enkripsi dan dekripsi. Parameter yang dibangkitkan adalah :
   a. Bilangan prima *q* yang dipilih secara acak dengan jumlah bit tertentu
   b. *group G1, G2* dengan *order q* bersama dengan bilinear map $e : G_1 \times G_1 \rightarrow G_2$, yang memenuhi persyaratan untuk menciptakan *BDHP*
   c. Elemen *P* yang dipilih secara acak dari *group* G_1 ($P \in G_1$)
   d. Bilangan acak $s \in Z$ dan nilai $P_{Pub} = sP$
   e. Fungsi hash :
      i. $H_1 : \{0,1\}^* \rightarrow G_1^*$
      ii. $H_2 : G_2 \rightarrow \{0,1\}^n$

   Plaintext direpresentasikan sebagai $\mathcal{M} = \{0,1\}^n$ dan *ciphertext* direpresentasikan sebagai $\mathcal{C} = G_1 \times \{0,1\}^n$. Parameter sistem yang bersifat publik adalah $\langle q, G_1, G_2, e, P, P_{Pub}, H_1, H_2 \rangle$, sedangkan nilai *s* hanya diketahui oleh PKG (*master key*).
2. *Extract*, proses untuk mendapatkan kunci privat suatu entitas. Diberikan suatu nilai *string* ID $\in \{0,1\}^*$, sistem akan menghitung $Q_{ID} = H_1(ID) \in G_1^*$, kemudian ditentukan $d_{ID} = sQ_{ID}$, di mana *s* adalah *master key*. Proses ini hanya dapat dilakukan oleh *PKG.*
3. *Encrypt*, proses mengubah *plaintext* $M \in \mathcal{M}$ menjadi *ciphertext* $C \in \mathcal{C}$ dengan menggunakan kunci publik ID. Proses ini dapat dilakukan oleh tiap entitas yang telah mengetahui nilai – nilai parameter sistem yang dihasilkan pada proses *Setup*. Langkah – langkah yang dilakukan adalah sebagai berikut :
   a) hitung $Q_{ID} = H_1(ID) \in G_1^*$
   b) pilih bilangan acak $r \in Z_q^*$
   c) *ciphertext* ditentukan dengan menghitung nilai :

   $$C = \langle U, V \rangle = \langle rP, M \oplus H_2(g_{ID}^r) \rangle$$

   dengan $g_{ID} = e(Q_{ID}, P_{Pub})$
4. *Decrypt*, proses mengubah *ciphertext* menjadi *plaintext*. Untuk melakukan proses ini sebuah entitas harus terlebih dahulu mendapatkan kunci privat d_{ID} dari *PKG* melalui proses *Extract*. *Ciphertext* $C \in \mathcal{C}$ berbentuk *tuple* $\langle U, V \rangle$. Perhitungan nilai $M \in \mathcal{M}$ dari *C* adalah sebagai berikut :

   $$V \oplus H_2(e(d_{ID}, U)) = M$$

Skema BF-IBE tidak mencakup pembangkitan tanda tangan digital untuk suatu pesan tertentu, karena itu diperlukan algoritma tanda tangan digital berdasarkan BF-IBE. Untuk melakukan efisiensi sistem, mekanisme *digital signature* harus memanfaatkan parameter yang digunakan pada skema *BF-IBE* yaitu parameter sistem, kunci publik dan kunci privat. Untuk pemberian tanda tangan digital dan verifikasinya, dibutuhkan fungsi *hash* $H_3 : \{0,1\}^* \rightarrow F_q$. Parameter sistem keseluruhan menjadi : $\langle q, G_1, G_2, e, P, P_{Pub}, H_1, H_2, H_3, E, D \rangle$. Kunci privat dinotasikan dengan d_{ID} dan kunci publik dinotasikan dengan Q_{ID}. Pesan yang akan diberikan tanda tangan digital dinotasikan dengan *m*. Berikut mekanisme pemberian tanda tangan digital dan verifikasinya :

1. Pemberian Tanda Tangan Digital
   a) Pilih nilai acak $t \in F_q$
   b) Pilih elemen acak $P_1 \in G_1$
   c) Hitung nilai $r = e(P_1, P)^t$
   d) Hitung nilai h = $H_3(m \mid\mid r)$, operator $\mid\mid$ menyatakan operator *concat* (penggabungan *string*)
   e) Hitung nilai $W = hd_{ID} + tP_1$
   f) Tanda tangan digital pada pesan *m* adalah pasangan nilai $\langle W, h \rangle$
2. Verifikasi Tanda Tangan Digital
   a) Hitung nilai $r = e(W,P) \bullet e(Q_{ID}, -P_{Pub})^h$
   b) Tanda tangan $\langle W, h \rangle$ pada pesan m dinyatakan valid jika dan hanya jika :

      $$h = H_3(m \mid\mid r)$$

Pada skema di atas, pemberian tanda tangan digital memerlukan kunci privat pengirim yang akan memberikan tanda tangan digital dan bergantung pada isi pesan (*m*). Sedangkan verifikasi hanya memerlukan parameter sistem, kunci publik pengirim pesan dan isi pesan (*m*) sehingga dapat dilakukan oleh siapa saja yang ingin memverifikasi isi pesan dan pengirim pesan.

4. Implementasi IBC

Implementasi IBC terbagi atas 2 aplikasi yaitu aplikasi *plug-in* untuk *Microsoft Outlook* dan aplikasi *web* yang berfungsi sebagai PKG. Proses analisis, perancangan dan implementasi

menggunakan skema *Object Oriented Analysis and Design (OOAD)*. Kelas - kelas yang diimplementasikan untuk masing – masing aplikasi adalah sebagai berikut :

1. Outlook Connector (aplikasi *plug-in*) bertanggung jawab memberikan akses terhadap objek – objek *Outlook*
2. IBCSystemParameter (aplikasi *plug-in* dan aplikasi *web*) bertanggung jawab memproses arsip yang berisi parameter sistem dan menyimpan semua parameter sistem dalam arsip.
3. IBCKey (aplikasi *plug-in*) bertanggung jawab menentukan nilai kunci publik dan menentukan nilai kunci privat
4. Enkripsi/Dekripsi Simetri (aplikasi *plug-in* dan aplikasi *web*) bertanggung jawab melakukan proses enkripsi kunci simetri dan melakukan dekripsi kunci simetri
5. Enkripsi/Dekripsi IBC (aplikasi *plug-in*) bertanggung jawab melakukan proses enkripsi dengan skema *BF-IBE* dan melakukan proses dekripsi dengan skema *BF-IBE*
6. DigitalSinature IBC (aplikasi *plug-in*) bertanggung jawab memberikan *digital signature* pada *email* sesuai isi dan pengirim *email* serta verifikasi terhadap *digital signature*
7. WebUI (aplikasi *web*) bertanggung jawab memberikan *interface* input bagi pengguna.
8. IBCLibrary (aplikasi *web*) bertanggung jawab membangkitkan parameter sistem dan membangkitkan kunci privat.

Dalam implementasi IBC, digunakan dua *library* eksternal yaitu :

1. **IdentityBasedEncryption JCA. 1.0.4,** *library* ini digunakan untuk melakukan operasi matematis pada kurva elliptik, *finite field* dan *bilinear mapping*. Implementasinya terdapat dalam arsip **`IBELibrary.dll`.** *Library* ini dikembangkan oleh *Computer Security and Cryptography Group, Computer Science Department, National University of Ireland Maynooth* (www.crypto.cs.may.ie)
2. **Bouncy Castle Cryptographic C# API**, *library* ini *framework* digunakan untuk melakukan operasi dasar kriptografi seperti *hash function*, *digital signature* serta implementasi tipe bilangan bulat yang besar *(BigInteger)*. Implementasinya terdapat dalam arsip `CryptoAPI.dll`. *Library* ini dikembangkan oleh *The Legion Of The Bouncy Castle* (www.bouncycastle.org)

Lingkungan perangkat keras tempat pembangunan perangkat lunak ini adalah sebagai berikut:

1. Prosesor Pentium 4 1.5 GHz.
2. Memori 512 MB
3. VGA true color.

Sedangkan lingkungan perangkat lunaknya dijelaskan sebagai berikut:

1. Sistem operasi *Windows XP Profesional Edition.*
2. Bahasa pemrograman C# dengan *tool Microsoft Visual Studio .NET 2003*.
3. Bahasa pemrograman *web ASP.Net*
4. *Microsoft Outlook 2003*
5. *Webserver Internet Information Server 5.1*

Alasan mengapa perangkat lunak ini dikembangkan di atas sistem operasi *Windows XP*, karena sistem operasi ini kini telah dipakai secara umum oleh banyak orang di seluruh dunia, dengan begitu diharapkan perangkat lunak ini dapat digunakan hampir oleh seluruh orang. Sedangkan *C#* dan *Microsoft Visual Studio .NET 2003* digunakan karena kebutuhan untuk pemrograman berorientasi objek serta kemudahan yang ditawarkan dalam membangun *Outlook plug-in*. Pemilihan *ASP.Net* didasarkan pada kompatibilitasnya dengan *C#*, sehingga kelas-kelas pada *Outlook plug-in* dapat digunakan pada pembangunan aplikasi *web*.

Pada proses pengujian perangkat lunak ini didapatkan hasil bahwa perangkat lunak telah mampu memberikan pengamanan baik pada *content* dan *attachment email*. Tingkat keamanan yang diberikan mencakup otentikasi, *confidentiality* dan *data integrity* melalui proses enkripsi-dekripsi serta *non repudiation* melalui proses pemberian tanda tangan digital-verifikasi.

5. Kesimpulan

Kesimpulan yang dapat diambil adalah :

1. *Identity Based Cryptography (IBC)* merupakan sebuah solusi untuk melakukan pengamanan data dengan sistem kunci publik tanpa harus menyulitkan penggunanya dalam melakukan manajemen kunci. Aplikasi IBC juga dapat digunakan sebagai langkah awal dalam penggunaan sistem kriptografi kunci publik.
2. Penggunaan IBC merupakan salah satu solusi yang tepat untuk keamanan komunikasi *email* karena adanya informasi yang berlaku umum dan unik yang dapat digunakan sebagai pembangkit pasangan kunci privat dan kunci publik yaitu alamat *email*.
3. Hasil implementasi IBC dengan menggunakan *outlook plug-in* memberikan kemudahan bagi pengguna *email* khususnya yang menggunakan *Microsoft Outlook* sebagai *email client*. Hal ini dikarenakan penggunaan *plug-in* pada *email client* yang telah biasa digunakan oleh pengguna.
4. Skema proses enkripsi dan dekripsi pada BF-IBE dapat dikembangkan untuk memberikan tanda tangan digital dan proses verifikasinya. Dengan pengembangan skema tersebut, IBC menjadi sistem kriptografi yang lengkap dan mampu memenuhi *security requirement* secara utuh.

6. Daftar Pustaka

1. Baldwin M. *Identity Based Encryption from the TatePairing to Secure Email Communications . University of Bristol*. 2002.
2. Boneh, D. dan Franklin, M. *Identity-Based Encryption from the Weil Pairing. Advances in Cryptology*. Crypto 2001. *URL :* <http://crypto.stanford.edu/~dabo/papers/ibe.pdf >. *Tanggal akses : 5 Februari 2005*
3. Duffy, Adam and Dowling, Tom. *An Object Oriented Approach to an Identity Based Encryption. Computer Security and Cryptography Group, Computer Science Department, National University of Ireland, Maynooth*. 2003.
4. Menezes, A.J., Van Oorschot, P.C. Vanstone, S.A. *Handbook of Applied Cryptography*. CRC Press. 1997.
5. Schneier, Bruce. *Applied Cryptography, 2nd edition*. McGraw-Hill. 1996.
6. Shamir, A. *Identity-based cryptosystems and signature schemes. Advances in Cryptology. Crypto '84*, Lecture Notes in Computer Science, Vol. 196, Springer-Verlag, pp. 47-53. 1984.