Jurnal Bisantara Informatika (JBI)
Vol.3, No.1, Juni 2019, pp. 1~17
ISSN (print): 2686-6455, ISSN (online): 2686-5319

KEAMANAN DATA DENGAN SKEMA *Multi-Key Hierarchical Identity-Based Signature*

Allwine Allwine[1], Jimmi Hendrik P. Sitorus[2]
[1]STMIK Methodist Binjai; Jl. Jend. Sudirman No. 136 Binjai, 061 - 88742021 ,Indonesia
[2]AMIK Parbina Nusantara; Jl. Pane No. 34 Pematangsiantar, 0622-434084, Indonesia
[1]Teknik Informatika, STMIK Methodist Binjai, Binjai, Indonesia
[2]Teknik Informatika, AMIK Parbina Nusantara, Pematangsiantar, Indonesia
e-mail: [1]allwin@stmikmethodistbinjai.ac.id, [2]jimmisitorus@amikparbinanusantara.ac.id

Abstrak

Hoon Wei Lim dan Kenneth G. Paterson memperkenalkan sebuah primitif kriptografi baru yang disebut sebagai Multi-Key Hierarchical Identity-Based Signatures (multi-key HIBS). Pada Hierarchical Identity-Based Cryptography (HIBC), multiple level dari generator kunci privat (private key generator / PKG) dan user membentuk sebuah struktur mirip pohon yang meniru model hirarki PKI. Seorang user pada sembarang level pada pohon dapat mengenkripsi atau menandatangani sebuah pesan untuk penerima yang diinginkan pada level tertentu, hanya menggunakan sekumpulan parameter sistem kriptografi yang digunakan bersama-sama yang dipublikasikan oleh akar PKG. Sebuah skema hierarchical identity-based signature (HIBS) merupakan analog dari sebuah pengaturan hirarki pada IBS. Pada hierarchical identity-based cryptography (HIBC), seorang user pada sembarang level pada pohon dapat mengenkripsi atau menandatangani sebuah pesan untuk penerima yang diinginkan pada level tertentu, hanya menggunakan sekumpulan parameter sistem kriptografi yang digunakan bersama-sama yang dipublikasikan oleh akar PKG. Sebuah skema hierarchical identity-based signature (HIBS) ini dapat digunakan untuk pendelegasian secara alami. Sebuah bukti pendelegasian dari utusan (delegatee) dapat dicek dengan hanya melakukan verifikasi satu signature, tidak tergantung panjang dari rantai delegasi. Penelitian ini merupakan implementasi Multi key HIBS skema Hoon Wei Lim dan Kenneth G. Paterson. Perangkat lunak yang dibuat dapat digunakan sebagai media pemahaman skema tersebut dan aplikasi yang dibuat telah memenuhi persyaratan skema sehingga dapat diaplikasikan dalam kehidupan nyata.

Kata kunci*— Multi-Key Hierarchical Identity-Based Signatures (multi-key HIBS), Kriptografi, Hierarchical Identity-Based Cryptography (HIBC)*

Abstract

Hoon Wei Lim and Kenneth G. Paterson introduced a new cryptographic primitive called multi-key hierarchical identity-based signatures (multi-key HIBS). In hierarchical identity-based cryptography (HIBC), multiple levels of the private key generator (PKG) and user form a tree-like structure that mimics the PKI hierarchy model. A user on any level in a tree can encrypt or sign a message for the desired recipient at a certain level, using only a set of cryptographic system parameters used together which are published by the PKG root. A hierarchical identity-based signature (HIBS) scheme is analogous to a hierarchical setting on IBS. In hierarchical identity-based cryptography (HIBC), a user at any level in a tree can encrypt or sign a message for the desired recipient at a certain level, using only a set of cryptographic system parameters that are used together which are published by PKG roots. A hierarchical identity-based signature (HIBS) scheme can be used for natural delegation. A proof of delegation from delegates can be checked by only verifying one signature, not

depending on the length of the delegation chain. This research is the implementation of the Multi key HIBS scheme Hoon Wei Lim and Kenneth G. Paterson. The software created can be used as a medium for understanding the scheme and applications that are made have met the requirements of the scheme so that it can be applied in real life.

Keywords— *Multi-Key Hierarchical Identity-Based Signatures (multi-key HIBS), Cryptography, Hierarchical Identity-Based Cryptography (HIBC)*

1. PENDAHULUAN

Penelitian pada *identity-based cryptography (IBC)* telah berkembang dalam beberapa tahuan terakhir ini sejak penemuan dari protokol persetujuan kunci berbasis *pairing* (*pairing based kay agreeement protocol)* yang dilakukan oleh Sakai dan kompatriot dan Joux dan pekerjaan dari Boneh dan Franklin memberikan skema enkripsi berbasis identitas *(identity-based encryption/IBE)* perta yang aman dan praktikal. Beberapa proposal untuk skem *identity-based signature (IBS)* dan diikuti oleh publikasi Boneh dan Franklin. Walaupun demikian, skema IBS kurang menarik jika dibandingkan dengan skema IBE. Skema IBE tersebut memiliki kelemahan dari sifat *key escrow escrow* yang di turunkan dari IBC, yang membuat sifat ketidakmampuan penyangkalan (*non-reputation*) dari *signature* ini menjadi sangat sudah dicapai. Lebih lanjut lagi, sebuah konstruksi generik yang menghasilkan skema IBS dari dua instansiasi dari sembarang skema *signature* kunci publik normal. Hal ini berhubungan dengan fakta bahwa dengan sebuah *siganture* berbasis sertifikat normal, semua informasi yang diperlukan untuk verifikasi dapat di kelompokkan dengan signature dasar untuk menghasilkan sebuah package. Pada *Hierarchical Identity-Based Cryptography (HIBC), multi level* dari generator kunci privat (*Private key generator/PKG*) dan *user* membentuk sebuah struktur mirip pohon yang meniru model hirarki *public key infrastructure (PKI)*. Seorang *user* pada sembarang *level* pada pohon dapat mengenkripsi atau menandatangani sebuah untuk penerima yang diinginkan pada level tertentu, hanya menggunakan sekumpulan parameter sistem kriptografi yang digunakan bersama-sama yang dipublikasikan oleh akar PKG. Sebuah skema *hierarchical identity-based signature (HIBS)* merupakan analog dari sebuah pengaturan hirarki pada IBS. Salah satu aplikasi menarik dari pendekatan hirarki yaitu skema ini dapat digunakan untuk pendelegasian secara alami. Sebuah bukti pendelegasian dari utusan (*delegatee*) dapat dicek dengan hanya melakukan verifikasi satu *signature*, tidak tergantung panjang dari rantai delegasi. Hoon Wei Lim dan Kenneth G. Paterson memperkenalkan sebuah primitif kriptografi baru yang disebut sebagai *multikey hierarchical identity-based signatures (multi key HIBS).* Hingga saat ini pendekatan ini belum diterapkan secara praktikal dalam kehidupan sehari-hari

2. METODE PENELITIAN

2.1 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Keamanan pesan diperoleh dengan menyandikannya menjadi pesan yang tidak mempuyai makna. Zaman sekarang, kerahasiaan mfonnasi menjadi sesuatu yang pentmg. Infonnasi yang rahasia perlu disembunyikan agar tidak diketahui oleh orang yang tidak berhak. Seseorang tentu tidak ingin nomor PIN kartu kredit atau kartu ATM- nya diketahui orang. Atau, jika suatu pesan ditulis secara rahasia dan tidak ingin diketahui atau dibaca oleh orang lain. Kriptografi bertujuan untuk memberikan layanan keamanan sepelti aspek-aspek keamanan Kerahasiaan, integritas data, Otentikasi, dan Nilpenyangkalan. (Munir, 2005, 9).

2.2 Data

Keterangan atau ilustrasi mengenai sesuatu hal bisa berbentuk kategori, misalnya rusak, baik, senang, puas, berhasil, gagal, dan sebagainya, atau bisa berbentuk bilangan. Semuanya ini dinamakan data. Data yang berbentuk bilangan disebut data kuantitatif, harganya belubah ubah atau bersifat variabel. Dari nilainya, dikenal dua golongan data kuantitatif yaitu,

a. Data dengan variabel diskrit atau singkatnya disebut data diskrit.
b. Data dengan variabel kontinu atau singkatnya disebut data kontinu.

Hasil menghitung atau membilang merupakan data diskrit eedangkan hasil pengukuran merupakan data kontinu. Contoh data diskrit yaitu,

a. Keluarga A mempunyai lima anak laki — laki dan tiga anak perempuan.
b. Kabupaten B sudah membangun 85 gedung sekolah.

Sedang contoh data kontinu dapat dilihat dalam tiga hal berikut,

a. Tinggi badan seseorang, misalnya 155 cm, 167 cm, atau 172,4 cm.
b. Luas daerah sebesar 425,7 km2.
c. Kecepatan mobll 60 km / jam.

Data yang bukan kuantitatif disebut data kualitatif. Ini tidak lain daripada data yang dikategorikan menurut lukisan kualitas objek yang dipelajari. Golongan ini dikenal pula dengan nama atribut. Misalnya sembuh, rusak, gagal, berhasil, dan sebagainya.

Menurut sumbemya, data terbagi dari dua macam yaitu:

1. Data Intern
   Pengusaha mencatat segala aktivitas perusahaannya sendiri: misalnya keadaan pegawai: pengeluaran: keadaan barang di gudang: hasil jualan: keadaan produksi pabriknya dan lain - lain aktivitas yang terjadl di dalam perusahan itu. Data yang diperoleh demikian merupakan data intern.
2. Data Ekstern
   Dalarn berbagai situasi: untuk perbandingan misalnya: diperlukan data dari sumber lain di luar perusahaan tadi. Data demiklan merupakan data ekstern. Data ekstern dibagi menjadi dua yaitu:
   a. Data Ekstern Primer atau dismgkat Data Primer, yaitu data yang dikeluarkan dan dikumpulkan oleh badan yang sarna.
   b. Data Ekstern Sekunder atau disingkat Data Sekunder: yaitu data yang dikeluarkan dan dikumpulkan oleh badan yang berbeda.

Data yang baru dikumpulkan belum pernah mengalami pengolahan apapun dikenal dengan nama data mentah. Suatu hal yang harus diperhatikan yaitu bagaimanapun dan darimanapun data diperoleh, kebenaran data yang diperoleh harus dapat di andalkan.

2. 3 Digital Signature

Digital Signature (tanda tangan digital) adalah istilah yang sering bermakna ambigu. Kadang, istilah ini diartikan sebagai bagian dari tanda tangan elektronik. Tetapi sebagian orang menggunakan istilah ini sebagai sesuatu yang sejajar dengan tanda tangan elektronik. *U.S Electronic Signatures in Global and National Commerce Act* yang diselenggarakan pada tahun 2000 menggunakan tanda tangan elektronik saat tanda tangan digital didiskusikan, mengilustrasikan kebingungan secara ilegal.

Tanda tangan digital berarti sebuah tanga tangan yang berbasiskan kepada skema kriptografi. Banyak yang sudah diajukan, beberapa sudah ditemukan dan sudah tidak ada lagi. Beberapa sudah dipatenkan, beberapa paten sudah tidak berlaku lagi, dan ada beberapa perbedaan pendapat dalam hal ini dengan tujuan komersil. (Munir, 2005, 17).

Tanda tangan digital ini menggunakan algoritma kanci nirsimetri dan biasanya menggunakan skema *Public Key Infrastructure (PKI)* dimana kunci publik yang digunakan dalam skema tanda tangan dengan pengguna oleh sebuah sertifikat identitas digital yang dikeluarkan oleh sebuah organisasi yang memiliki otoritas untuk mengeluarkan sertifikat, biasanya dikelola oleh perusahaan komersial pihak ketiga. Sistem PKI memiliki tujuan untuk

menyamarkan informasi ppengguna (nama, alamat, no telepon dan lain lain) menjadi sebuah kunci publik, ide dasarnya cukup dekat dengan tugas notaris. Ada beberapa skema tanda tangan yang umumnya menggunakan dua algoritma, satu untuk memberi tanda tangan dan yang lainnya untuk melakukan pengecekan keaslian tanda tangan tersebut. Keluaran dari proses tanda tangan disebut juga dengan tanda tangan digital.

2. 4 Algoritma Hash

Fungsi *Hash* mengurangi data dari ukuran yang berubah – ubah menjadi ukuran yang khusus. Fungsi Hash dibutuhkan dalam bagian konfigurasi sistem untuk memudahkan pengecekan terhadap kelebihan data. Seluruh data dapat diperiksa untuk melihat apakah data yang berkapasitas besar dapat diulang, sebab hal ini akan mendatangkan kerugian bessar dalam kecepatan dan waktu. (Bruce Schneier, 2007).

Fungsi Hash dapat digunakan dalam kriptografi, yaitu dalam hal membagi stribut yang mirip, terutama dalam hal tanda tangan digital. Sebagai contoh DD menandai 320 bit dari pesan 160 bit. Bagaimanapun juga ketika kalimat dapat lebih panjang, pesan ini akan menghasilkan kelambatan dalam proses pengiriman dan penyimpanan, karena panjang pppesan menjadi ganda dari pesan aslinya. Hal ini terjadi pada waktu sesudah tanda tangan dimasukkan dan dibagi dalam blok 16 bit untuk tanda tangan itu sendiri. Hal ini menyebabkan penurunan kecepatan dan pesan dianggap tidak valid. (Bruce Schneier, 2007).

Contoh:
X = {1,2,3,4,5}
Hash (x) = x mod 3
Maka:
Hash (1) = 1 Mod 3
Hash (1) = 2 Mod 3
Hash (1) = 3 Mod 3
Hash (1) = 4 Mod 3
Hash (1) = 5 Mod 3

2. 5 SHA (Secure Hash Algorithm)

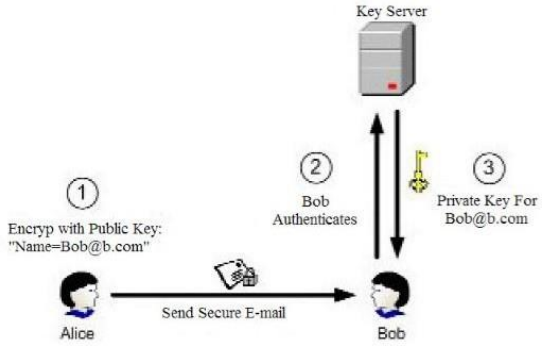
NIST bersama dengan NSA mendesain *Secure Hash Algorithm (SHA)* untuk digunakan sebagai komponen *Digital Signature Standard (DSS)*. Standar Hash adalah *Secure Hash Standar (SHS)* dengan SHA sebagai algoritma yang digunakan. Jadi, SHS adalah standar sedangkan SHA adalah algoritma.

Standar menetapkan *Secure Hash Algorithm (SHA)* yang diperlukan untuk menjamin keamanan *Digital Signature Algorithm (DSA)*. Ketika pesan dengan sembarang panjang $<2^{256}$ bit dimasukkan, SHA menghasilkan 160 bit keluaran ke dalam DSA, yang menghitung tanda tangan digital untuk pesan tersebut. Penandatanganan MD (dan bukannya penandatanganan pesan secara langsung) sering kali meningkatkan efisiensi proses, karena MD biasanya jauh lebih kecil dibandingkan pesan aslinya. MD pesan yang sama seharusnya dapat diperoleh dengan pemeriksa tanda tangan ketika menerima pesan dari pengirim dengan cara memasukkan pesan tersebut ke fungsi Hash SHA. SHA dikatakan aman karena didesain supaya secara metmatis tidak dimungkinkan untuk mendapatkan pesan aslinya bila diberikan hash-nya atau tidak mungkin mendapatkan dua pesan yang berbeda yang meghasilkan MD yang sama. SHA dibuat berdasarkan rancangan yang serupa dengan MD4 yang dibuat oleh Professor Ronald L. Rivest dari MIT. SHA menghasilkan keluaran sidik jari 160 bit, lebih panjang dibandingkan MD5. (Bruce Schneier, 2007).

2. 6 Identity Based Cripyosystem

Konsep *Identity-Based Cruptosystem* atai *Identity-Based Encryption(IBE)* ditemukan pada tahuan 1984 oleh Adi Shamir dalam rangka mengatasi masalah autentikasi kunci publik. Idenya adalah untuk menghindari kebutuhan autentikasi dengan cara kunci publik yang

digunakan berhubungan langsung dengan identitas user. Kunci publik user dihasilkan langsung dari informasi publik yang tersedia yang dapat mengidentifikasikan user tersebut secara unik. Informasi ini disebut sebagai 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 mengetahui identitasnya sehingga tidak diperlukan lagi pencarian kunci publik sehingga menghilangkan kebutuhan akan sertifikat seperti pada PKI. Bagaimanapun, realisasi hubungan antara user dengan identitas digitalnya cukup sulit.



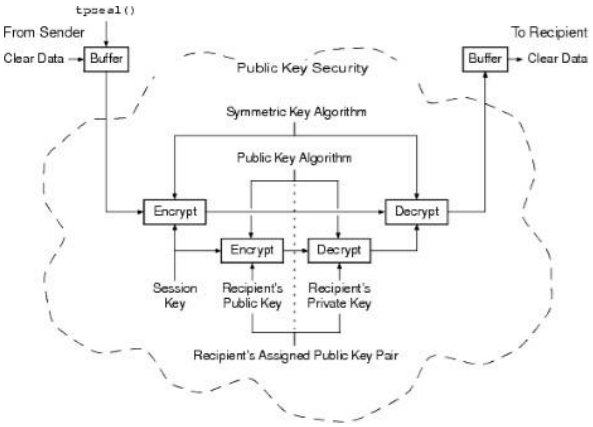
Gambar 1 Cara Kerja IBE

2. 7 Skema *Multi-Key Hierarchical Identity-Based Signature (HIBS)*

Pada Skema *Hierarchical Identity – Based Cryptography* ini, *Private Key Generator (PKG)* akan ditempatkan pada level 0 (akar). PKG ini berfungsi untuk mengasilkan kunci yang diperlukan oleh setiap user yang berada dibawahnya. Selain itu PKG juga akan menyimpan data identitas dari setiap user yang merupakan anak cabangnya.

Sebuah skema *MultiKey Hierarchical identity-based signature* (multi-key HIBS) dapat dianggap sebagai pengembangan dari skema HIBS. Skema ini menghasilkan *signature* pada pengaturan hirarki berbasis identitas dengan menggunakan sekumpulan kunci penandatanganan. Sebuah skema multi-key HIBS dapat dispesifikasikan oleh algoritma berikut:

1. *Root Setup*
   Algoritma ini dilakukan oleh PKG akar. Algoritma ini menghasilkan parameter sistem dan sebuah rahasia utama pada input sebuah parameter sekuritas λ. Parameter sistem, mencakup sebuah deskripsi
2. *Lower Level Setup*
   Semua entitas pada level lebih rendah harus memiliki parameter sistem yang dihasilkan oleh PKG akar. Algoritma ini memungkinkan sebuah PKG yang berlevel lebih rendah untuk menghasilkan sebuah nilai rahasia yang akan digunakan untuk menghasilkan kunci privat anaknya.
3. *Extract*
   Algoritma ini dijalankan oleh sebuah PKG(akar ataupun PKG yang berlevel lebih rendah) dengan identitas ID_t untuk menghasilkan sebuah kunci privat S_{t+1} untuk sembarang anaknya dengan menggunakan parameter sistem dan kunci privatnya.
4. *Sign*
   Diberikan sebuah set $SK = \{St_j^j : 1 \leq j \leq n\}$ dari kunci penandatanganan, sebuah pesan M dan parameter sistem, algoritma ini akan menghasilkan sebuah *signature* $\sigma \in S$. Disini t_j melambangkan level dari penandatanganan ke –j dalam kumpulan SK.
5. *Verify*
   Diberikan sebuah *signature* $\sigma \in S$ sebuah kumpulan $ID = \{Idt_j^j : 1 \leq j \leq n\}$ dari identitas, sebuah pesan M, dan parameter sistem, algoritma ini akan menghasilkan nilai valid atau tidak valid.



Gambar 2 *Hierarchy Identity Based Signature*

3. HASIL DAN PEMBAHASAN

2. 7 Perhitungan Skema HIBS

Berikut ini dideskripsikan sebuah skema konkrit dari HIBS *multi-key* yang diadaptasi dari skema HIBS Gentry-Silverberg.

a. Root Setup

Akar PKG menjalankan algoritma berikut:

1. Pilih nilai q = 17.
2. Pilih sebuah generator $P_0 = 11$.
3. Pilih sebuah nilai acak $s_0 = 5$
4. Hitunglah nilai Q_0

$Q_0 = s_0.P_0.$

$= 5 * 11$

$= 55$

Rahasia utama dari akar PKG adalah $s_0 = 5$

b. Lower – Level Setup

Misalkan t = 2, maka pilihlah nilai acak s_1 dan s_2:

$s_1 = 7.$

$s_2 = 11.$

c. Extract

tuk sebuah entitas dengan *identifier* $ID_t = id_1$ = 'A' dan id_2 = 'B', induk

itas melakukan proses berikut:

Hitung nilai:

P_1 = SHA-1(A)

= 01101101110011010100110011100010 (ambil 32 bit saja)

= 1842171106

Proses perhitungan :

H0 = 67452301
H1 = EFCDAB89
H2 = 98BADCFE
H3 = 10325476
H4 = C3D2E1F0

Pesan = 'A'
'A' = 65 = 01000001

Bit Pesan = 01000001
Dalam heksa = 41

Panjang pesan = 8 bit = 8 heksa
Tambahkan bit 1 ke bit pesan = 010000011

Tambahkan bit 0 ke pesan hingga panjangnya 448 bit =
dalam heksa =
41800000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

Tambahkan nilai panjang pesan pada akhir =
panjang pesan = 1000 bit = 8 heksa
Blok pesan dalam heksa =
41800000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000008

Bagi blok pesan ke dalam 16 buah subblok =
W(0) = 01000001100000000000000000000000 = 41800000 HEKSA
W(1) = 00000000000000000000000000000000 = 00000000 HEKSA
W(2) = 00000000000000000000000000000000 = 00000000 HEKSA
W(3) = 00000000000000000000000000000000 = 00000000 HEKSA
W(4) = 00000000000000000000000000000000 = 00000000 HEKSA
W(5) = 00000000000000000000000000000000 = 00000000 HEKSA
W(6) = 00000000000000000000000000000000 = 00000000 HEKSA
W(7) = 00000000000000000000000000000000 = 00000000 HEKSA
W(8) = 00000000000000000000000000000000 = 00000000 HEKSA
W(9) = 00000000000000000000000000000000 = 00000000 HEKSA
W(10) = 00000000000000000000000000000000 = 00000000 HEKSA
W(11) = 00000000000000000000000000000000 = 00000000 HEKSA
W(12) = 00000000000000000000000000000000 = 00000000 HEKSA
W(13) = 00000000000000000000000000000000 = 00000000 HEKSA
W(14) = 00000000000000000000000000000000 = 00000000 HEKSA
W(15) = 00000000000000000000000000001000 = 00000008 HEKSA

Hitung subblok 16 sampai 79 :
Rumus : W(i) = (W(i-3) xor w(i-8) xor W(i-14) xor W(i-16)) <<< 1

W(16) = (W(13) XOR W(2) XOR W(0)) <<< 1
W(16) = (00000000 XOR 00000000 XOR 00000000 XOR 41800000) <<< 1
W(16) = 83000000

DAN SETERUSNYA …

W(79) = (W(76) XOR W(65) XOR W(63)) <<< 1
W(79) = (400875F9 XOR 0014A518 XOR 00050000 XOR 80084701) <<< 1
W(79) = 80232FC1

Set nilai awal =
A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476
E = C3D2E1F0

Putaran ke - 0
---
f = (B AND C) OR (NOT(B) AND D)
f = (EFCDAB89 AND 98BADCFE) OR (NOT(EFCDAB89) AND 10325476)
f = 98BADCFE

Temp = (A <<< 5) + f + E + k + W(i)

Temp = (67452301 <<< 5) + 98BADCFE + C3D2E1F0 + 5A827999 + 41800000
Temp = E13498B3
E = 10325476
D = 98BADCFE
C = 7BF36AE2
B = 67452301
A = E13408B3

Putaran ke - 1
---
f = (B AND C) OR (NOT(B) AND D)
f = (67452301 AND 7BF36AE2) OR (NOT(67452301) AND 98BADCFE)
f = FBFBFEFE

Temp = (A <<< 5) + f + E + k + W(i)
Temp = (E13408B3 <<< 5) + FBFBFEFE + 10325476 + 5A827999 + 00000000
Temp = 8D43E389
E = 98BADCFE
D = 7BF36AE2
C = 59D148C0
B = E13498B3
A = 8D43E389

Dan seterusnya hingga diperoleh hasil sebagai berikut:

H0 = H0 + A = 67452301 + 068829E1 = 6DCD4CE2
H1 = H1 + B = EFCDAB89 + 4DBB3765 = 3D88E2EE
H2 = H2 + C = 98BADCFE + FCADDD56 = 9568BA54
H3 = H3 + D = 10325476 + 5BCE27ED = 6C007C63
H4 = H4 + E = C3D2E1F0 + 15403A2B = D9131C1B

Output (ambil 32 bit saja) = 01101101110011010100110011100010

P_2 = SHA-1(B)

= 10101110010011110010100000011101 (ambil 32 bit saja)

= 2924423197

Proses perhitungan :

H0 = 67452301
H1 = EFCDAB89
H2 = 98BADCFE
H3 = 10325476
H4 = C3D2E1F0

Pesan = 'B'
'B' = 66 = 01000010

Bit Pesan = 01000010
Dalam heksa = 42

Panjang pesan = 8 bit = 8 heksa
Tambahkan bit 1 ke bit pesan = 010000101

Tambahkan bit 0 ke pesan hingga panjangnya 448 bit =
dalam heksa =
428000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

Tambahkan nilai panjang pesan pada akhir =
panjang pesan = 1000 bit = 8 heksa
Blok pesan dalam heksa =
428000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000008

Bagi blok pesan ke dalam 16 buah subblok =
W(0) = 01000010100000000000000000000000 = 42800000 HEKSA
W(1) = 00000000000000000000000000000000 = 00000000 HEKSA
W(2) = 00000000000000000000000000000000 = 00000000 HEKSA
W(3) = 00000000000000000000000000000000 = 00000000 HEKSA
W(4) = 00000000000000000000000000000000 = 00000000 HEKSA
W(5) = 00000000000000000000000000000000 = 00000000 HEKSA
W(6) = 00000000000000000000000000000000 = 00000000 HEKSA
W(7) = 00000000000000000000000000000000 = 00000000 HEKSA
W(8) = 00000000000000000000000000000000 = 00000000 HEKSA
W(9) = 00000000000000000000000000000000 = 00000000 HEKSA
W(10) = 00000000000000000000000000000000 = 00000000 HEKSA
W(11) = 00000000000000000000000000000000 = 00000000 HEKSA
W(12) = 00000000000000000000000000000000 = 00000000 HEKSA
W(13) = 00000000000000000000000000000000 = 00000000 HEKSA
W(14) = 00000000000000000000000000000000 = 00000000 HEKSA
W(15) = 00000000000000000000000000001000 = 00000008 HEKSA

Hitung subblok 16 sampai 79 :
Rumus : W(i) = (W(i-3) xor w(i-8) xor W(i-14) xor W(i-16)) <<< 1

W(16) = (W(13) XOR W(2) XOR W(0)) <<< 1
W(16) = (00000000 XOR 00000000 XOR 00000000 XOR 42800000) <<< 1
W(16) = 85000000

DAN SETERUSNYA …

W(79) = (W(76) XOR W(65) XOR W(63)) <<< 1
W(79) = (C00816CB XOR 0014A628 XOR 00050000 XOR 80084702) <<< 1
W(79) = 8023EFC2

Set nilai awal =
A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476
E = C3D2E1F0

Putaran ke - 0
---
f = (B AND C) OR (NOT(B) AND D)
f = (EFCDAB89 AND 98BADCFE) OR (NOT(EFCDAB89) AND 10325476)
f = 98BADCFE

Temp = (A <<< 5) + f + E + k + W(i)
Temp = (67452301 <<< 5) + 98BADCFE + C3D2E1F0 + 5A827999 + 42800000
Temp = E23498B3
E = 10325476
D = 98BADCFE
C = 7BF36AE2
B = 67452301
A = E23498B3

Putaran ke - 1
---
f = (B AND C) OR (NOT(B) AND D)
f = (67452301 AND 7BF36AE2) OR (NOT(67452301) AND 98BADCFE)
f = FBFBFEF

Temp = (A <<< 5) + f + E + k + W(i)
Temp = (E23498B3 <<< 5) + FBFBFEFE + 10325476 + 5A827999 + 00000000
Temp = AD43E389
E = 98BADCFE
D = 7BF36AE2
C = 59D148C0
B = E23498B3
A = AD43E389

Dan seterusnya hingga diperoleh hasil sebagai berikut:

H0 = H0 + A = 67452301 + 470A051C = AE4F281D
H1 = H1 + B = EFCDAB89 + 05D82576 = F5A5D0FF
H2 = H2 + C = 98BADCFE + A3F28673 = 3CAD6371
H3 = H3 + D = 10325476 + E73B07B3 = F76D5C29
H4 = H4 + E = C3D2E1F0 + F30671FC = B6D953EC
Output (ambil 32 bit saja) = 10101110010011110010100000011101

2. Set nilai S_t

$S_t = \sum_{i=1}^{t} s_{i-1}P_i = S_{t-1} + s_{t-1}P_t$

$S_1 = s_0 * P_1$

$= 5 * 1842171106$

$= 9210855530$

$S_2 = S_1 + s_1 * P_2$

$= 9210855530 + 7 * 2924423197$

$= 29681817909$

3. Definisikan Q_i

$Q_i = s_i.P_0$ untuk $1 \leq i \leq t - 1$.

$Q_1 = s_1.P_0$

$= 7 * 11$

$= 77$

D. <u>Sign</u>

1. Pilih sebuah nilai rahasia $s_\varphi = 3$.
2. Pesan M = 'C'.
3. Hitung P_M

$P_M = H2(ID^l_{t1}, \ldots, ID^n_{tn}, M)$.

= SHA1('A', 'B', 'C').

= 00111100000000011011110110111011 (ambil 32 bit saja)

= 1006747067

Proses perhitungan:

H0 = 67452301
H1 = EFCDAB89

H2 = 98BADCFE
H3 = 10325476
H4 = C3D2E1F0

Pesan = 'ABC'
'A' = 65 = 01000001
'B' = 66 = 01000010
'C' = 67 = 01000011

Bit Pesan =
010000010100001001000011
Dalam heksa = 414243

Panjang pesan = 24 bit = 18 heksa
Tambahkan bit 1 ke bit pesan = 0100000101000010010000111

Tambahkan bit 0 ke pesan hingga panjangnya 448 bit =
dalam heksa =
414243800000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000

Tambahkan nilai panjang pesan pada akhir =
panjang pesan = 00011000 bit = 18 heksa
Blok pesan dalam heksa =
414243800000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
00000018

Bagi blok pesan ke dalam 16 buah subblok =
W(0) = 01000001010000100100001110000000 = 41424380 HEKSA
W(1) = 00000000000000000000000000000000 = 00000000 HEKSA
W(2) = 00000000000000000000000000000000 = 00000000 HEKSA
W(3) = 00000000000000000000000000000000 = 00000000 HEKSA
W(4) = 00000000000000000000000000000000 = 00000000 HEKSA
W(5) = 00000000000000000000000000000000 = 00000000 HEKSA
W(6) = 00000000000000000000000000000000 = 00000000 HEKSA
W(7) = 00000000000000000000000000000000 = 00000000 HEKSA
W(8) = 00000000000000000000000000000000 = 00000000 HEKSA
W(9) = 00000000000000000000000000000000 = 00000000 HEKSA
W(10) = 00000000000000000000000000000000 = 00000000 HEKSA
W(11) = 00000000000000000000000000000000 = 00000000 HEKSA
W(12) = 00000000000000000000000000000000 = 00000000 HEKSA
W(13) = 00000000000000000000000000000000 = 00000000 HEKSA
W(14) = 00000000000000000000000000000000 = 00000000 HEKSA
W(15) = 00000000000000000000000000011000 = 00000018 HEKSA

Hitung subblok 16 sampai 79 :

Rumus : W(i) = (W(i-3) xor w(i-8) xor W(i-14) xor W(i-16)) <<< 1

W(16) = (W(13) XOR W(2) XOR W(0)) <<< 1
W(16) = (00000000 XOR 00000000 XOR 00000000 XOR 41424380) <<< 1
W(16) = 82848700

DAN SETERUSNYA …

W(79) = (W(76) XOR W(65) XOR W(63)) <<< 1
W(79) = (F4D22D7F XOR 678564D6 XOR 000F0000 XOR 425B4981) <<< 1
W(79) = A2060051

Set nilai awal =
A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476
E = C3D2E1F0

Putaran ke - 0
---
f = (B AND C) OR (NOT(B) AND D)
f = (EFCDAB89 AND 98BADCFE) OR (NOT(EFCDAB89) AND 10325476)
f = 98BADCFE

Temp = (A <<< 5) + f + E + k + W(i)
Temp = (67452301 <<< 5) + 98BADCFE + C3D2E1F0 + 5A827999 + 41424380
Temp = E0F6DC33
E = 10325476
D = 98BADCFE
C = 7BF36AE2
B = 67452301
A = E0F6DC33

Putaran ke - 1
---
f = (B AND C) OR (NOT(B) AND D)
f = (67452301 AND 7BF36AE2) OR (NOT(67452301) AND 98BADCFE)
f = FBFBFEFE

Temp = (A <<< 5) + f + E + k + W(i)
Temp = (E0F6DC33 <<< 5) + FBFBFEFE + 10325476 + 5A827999 + 00000000

Temp = 858C5389
E = 98BADCFE
D = 7BF36AE2
C = 59D148C0
B = E0F6DC33
A = 858C5389

Dan seterusnya hingga diperoleh nilai sebagai berikut:

H0 = H0 + A = 67452301 + D4BC9ABA = 3C01BDBB
H1 = H1 + B = EFCDAB89 + 3725AD31 = 26F358BA
H2 = H2 + C = 98BADCFE + 19C4497B = B27F2679
H3 = H3 + D = 10325476 + 1477D824 = 24AA2C9A
H4 = H4 + E = C3D2E1F0 + 402A1BC8 = 03FCFDB8

Output (ambil 32 bit saja) : 00111100000000011011110110111011

Hitung nilai berikut:

$$\varphi = \sum_{j=1}^{n} S_{t_j}^{j} + s_{\varphi} P_M$$

$\varphi = S_1 + s_1 * P_M + S_2 + s_2 * P_M$

$\varphi = 9210855530 + 7 * 1006747067 + 29681817909 + 11 * 1006747067$

$\varphi = 818287643256$

$Q_\varphi = s_\varphi * P_0$
$= 3 * 11$
$= 33$

E. Verify

1. Hitung $P^j_i = H_1(ID^j_i)$ untuk $1 \leq i \leq t_j$ dan $1 \leq j \leq n$.

   $P_1 = SHA\text{-}1(A)$
   $= 01101101110011010100110011100010$ (ambil 32 bit saja)
   $= 1842171106$

P_2 = SHA-1(B)

= 10101110010011110010100000011101 (ambil 32 bit saja)

= 2924423197

2. Hitung nilai $P_M = H2(ID^l_{t1}, \ldots ID^l_{tn}, M)$

$P_M = H2(ID^l_{t1}, \ldots, ID^n_{tn}, M)$.

= SHA1('A', 'B', 'C').

= 00111100000000011011110110111011 (ambil 32 bit saja)

= 1006747067

3. Cek apakah persaman $e(P_0, \varphi)$ berikut terpenuhi:

$e(P0, \varphi) = 189$

Mul e = 21

$e(Q\varphi, P_M) = 9$

Proses verifikasi sukses !

4. KESIMPULAN

Setelah melaksanakan penelitian ini, maka dapat diambil kesimpulan bahwa, sebuah dokumen yang telah ditandatangani dapat diverifikasi dengan valid dengan menggunakan skema *Multi-Key HIBS*, dan juga perangkat lunak yang di desain memberikan fitur yang sederhana sehingga para pemakai dapat dengan mudah menggunakan fasilitas yang terdapat dalam perangkat lunak baik itu dibagian aplikasi maupun bagian Pemahaman.

5. SARAN

Bagian ini adalah opsional. Apabila ada maka saran-saran berisi saran penelitian lebih lanjut untuk menutup kekurangan penelitian saat ini atau pengembangan dari penelitian yang sudah dilakukan.

DAFTAR PUSTAKA

[1] Hoon Wei Lim dan Kenneth G. Paterson (2007). ***Multi-key Hierarchical Identity-Based Signatures***, Information Security Group, University of London.

[2] Liming, F. (2008). ***Full Security : Fuzzy Identity Based Encryption***. http://eprint.iacr.org/2008/307.pdf. Tanggal Akses 15 Maret 2012

[3] Piyi, Y et al. (2008). ***Fuzzy Identity Based Signature***. http://eprint.iacr.org/2008/ 002.pdf. Tanggal Akses 11 Maret 2012

[4] Munir R. (2008). **Pengantar Kriptografi**. http://www.informatika.org/ ~rinaldi/Buku/Kriptografi/Bab-1_Pengantar%20Kriptografi.pdf. Tanggal Akses 15 Maret 2012.

[5] Munir, R. (2006). ***Kriptografi***, Penerbit Informatika Bandung.

[6] Roger S. Pressman, Ph.D. (2002). **Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku Satu)**, Mc Graw-Hill Companies, Inc, Penerbit ANDI.

[7] Sodhi, J. (1991). ***Software Engineering Methods, Management, and CASE Tools***, TAB Professional dan Reference Books, Amerika.