Studi dan Implementasi RSA, SHA-1, TimeStamp Untuk penangangan Non Repudiation

Ecko Fernando Manalu 13508604
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18604@students.if.itb.ac.id

***Abstract*— RSA selain baik untuk digunakan dalam mengenkripsi dan mendekripsi dokumen dapat digunakan juga untuk menangani non-repudiation attack atau penyangkalan.**
Karakteristik RSA yakni menggunakan kunci asimetrik (yang tidak sama) ketika mengenkripsi dan mendekripsi, membuat RSA baik digunakan dalam penanganan penyangkalan, karena pasangan kunci publik dan privatnya adalah selalu unik, seperti pemetaan satu-ke-satu (one-to-one correspondence).

Bagaimana hal ini dimungkinkan adalah sebagai berikut:
Kita mengetahui bahwa Kriptografi dengan Kunci Asimetrik menggunakan pasangan kunci yang berbeda satu sama lain. Disebut kunci publik dan kunci privat. Ketika suatu pesan dienkrip dengan menggunakan kunci publik penerima, maka pesan tersebut hanya dapat dibuka dengan menggunakan kunci privat penerima. Hal sebaliknya sebenarnya berlaku, ketika suatu pesan dienkripsi dengan kunci privat penerima, maka hanya dapat dibuka dengan kunci publik si penerima juga. Hal tersebut benar dapat terjadi secara matematis. (Pembuktian dibahas lebih dalam di makalah).

RSA untuk non repudiation attack ini, dapat digabung dengan SHA-1, dan ditambahkan dengan Time Stamp (Penanda Waktu). Time Stamp adalah suatu identitas yang menandakan ke-validan suatu pesan berdasarkan periode antara waktu pengiriman dan waktu tiba suatu pesan di sisi penerima.

Studi Penggunaan RSA dan Time Stamp untuk Penanganan Penyangkalan ini akan dicoba untuk digabungkan dengan manajemen kunci sehingga diharapkan nantinya dapat memberikan suatu kontribusi baru untuk Kriptografi.

Kata Kunci: ***RSA, Non Repudiation, Time Stamp, SHA-1, Penyangkalan***

I. PENDAHULUAN

Latar Belakang

Dalam dunia internet tidak dapat dipungkiri bahwa transaksi akan acapkali terjadi. Segala bentuk transaksi tersebut terdiri dari banyak jenis dan tingkat kepentingan dan tingkat kerahasiaan yang tinggi. Ada transaksi yang sesederhana hanya layaknya sebuah obrolan singkat (chatting), atau bahkan ada yang sampai kepada transaksi yang melibatkan sejumlah besar uang. Sekecil apapun nilai transaksi tersebut, sudah suatu keharusan bahwa transaksi tersebut harus dilindungi, demi melindungi masing-masing pihak yang melakukan transaksi.

Pada sebuah contoh kasus ketika seseorang, kita sebut Antasari ingin mengirimkan pesan transaksi kepada seorang lainnya kita sebut Bibit. Antasari mengirim pesan agar Bibit menyetor uang sejumlah 10 Milyar kepada Chandra. Dalam pengiriman pesan, bisa saja ada seorang kita sebut Bob, menyadap pesan tersebut dan mengubah pesan. Bisa saja Bob mengubah pesan menjadi menyetor uang sejumlah 100 Milyar kepada Anggodo, atau bentuk lainnya. Lebih jauh lagi, bisa saja Antasari mengatakan bahwa yang mengirim pesan itu bukan dirinya tapi adalah Bob.

Kasus di atas termasuk kepada kasus penyangkalan transaksi. Jika dicoba untuk dipilah lebih detil lagi, akan banyak spekulasi yang timbul dari persoalan di atas, diantaranya :

1. Antasari bisa saja memang benar-benar tidak mengirimkan pesan tersebut, atau
2. Bisa saja dia memang mengirim tapi bukan sejumlah tersebut (diubah oleh Bob di tengah proses pengiriman), atau
3. Bisa saja dia memang berniat untuk menipu Bibit.

Dari pemaparan persoalan di atas dapat ditarik kesimpulan bahwa dibutuhkan suatu system yang dapat menangani atau mencegah terjadinya konflik akibat persoalan di atas. Kriptografi merupakan satu cabang ilmu informatika yang menangani masalah keamanan data termasuk kasus penyangkalan di atas.

Makalah ini akan mencoba memberi suatu solusi atas permasalahan tersebut di atas dan mencoba untuk memberi usulan implementasi meski hanya prototip fungsi, dan lebih lanjut lagi memberikan suatu arahan untuk pengembangan atau penelitian lebih lanjut.

Rumusan Masalah

Dari pemaparan mengenai persoalan penyangkalan di atas, dapat disimpulkan bahwa ada beberapa hal yang jadi rumusan persoalan yang akan dibahas dalam makalah ini:

1. Bagaimana membuat suatu mekanisme transaksi yang dapat memastikan (autentikasi) bahwa yang mengirim pesan adalah pengirim yang sebenarnya.
2. Bagaimana memastikan bahwa data tidak dapat dilihat oleh pihak yang tidak berkepentingan.
3. Bagaimana membuat suatu mekanisme transaksi yang dapat memastikan bahwa tidak terjadi pengubahan pesan di tengah pengiriman transaksi.
4. Bagaimana memastikan bahwa memang pesan tersebut jadi dikirim.

Pendekatan Penyelesaian

Untuk menyelesaikan persoalan yang telah dipaparkan di rumusan masalah di atas, secara garis besar cara yang digunakan adalah :

1. Menggunakan mekanisme kriptografi kunci public dan privat dari RSA untuk memastikan bahwa pesan akan terenkripsi sehingga tidak dapat dibaca oleh pihak lain.
2. Menggunakan mekanisme digest pesan dengan SHA-1 untuk menciptakan digital signature untuk memastikan bahwa pesan tidak akan diubah di tengah pengiriman.
3. Menggunakan mekanisme enkripsi digest + SHA-1 untuk menangani autentikasi.
4. Menggunakan timestamp untuk memastikan bahwa pesan memang dikirim, dan tidak diubah di tengah transaksi dengan membandingkan waktu pengiriman dokumen dengan waktu pengecapan (stamping) di pihak ketiga yang terpercaya disebut (TSA / Time Stamp Authority).

Ruang Lingkup dan Batasan

Dalam penyelesaian makalah, yang menjadi ruang lingkup dan batasan pembahasan adalah:

1. Hanya menangani terutama kepada kasus non-repudiation
2. Tidak menangani masalah apabila pengguna system tercuri password atau info berkaitan dengan akun-nya
3. Untuk Time-stamping, meski terdiri dari banyak mekanisme, diserahkan kepada implementor untuk memilih mekanisme mana yang lebih baik.

Sistematika Makalah

Makalah terdiri dari dua bagian utama, yakni :

1. Bagian mengenai studi bagaimana RSA, SHA-1 dan TSA dapat digunakan untuk menangani kasus non-repudiation
2. Bagian usulan implementasi mekanisme baru yang menggabungkan RSA, SHA-1 dan TSA untuk menangani kasus non-repudiation.

II. PEMBAHASAN

Kriptografi

Definisi lama menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dimengerti lagi maknanya [RIN01].

Kriptografi ini berkembang sehingga tidak lagi sebatas mengenkripsi pesan tapi juga memberikan aspek keamanan lainnya seperti :

1. Confidentiality (Kerahasiaan)
2. User Authentication (Otentikasi Pengguna)
3. Message Authentication (Otentikasi Pesan)
4. Non Repudiation (Penyangkalan)

Keempat aspek tersebut akan dicoba untuk diterapkan dalam mekanisme yang akan ditawarkan dalam makalah ini dengan pendekatan penggunaan RSA, SHA-1, Time Stamp.

RSA

RSA [RIN01, STA01, FAD01]adalah suatu algoritma kunci-publik yang dibuat oleh tiga orang peneliti dari MIT, Ron Rivest, Adi Shamir, Leonard Adleman. Kemanan algoritma ini telah diakui sebagai yang terbaik untuk saat ini, tentunya dengan banyak modifikasi dan gabungan dengan teknik lainnya.

Secara garis besar, keamanan algoritma RSA ini terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Bilangan yang dimaksud bervariasi ada yang besarnya 128 bit, hingga saat ini bahkan ada yang sudah mencapai 1KB. Proses pencarian bilangan tersebut juga sudah sangat sulit, dan proses penemuan factor primanya juga bahkan jauh lebih sulit, apalagi jika bilangan yang diambil sudah sangat besar.

Kecanggihan dari algoritma ini juga adalah keasimetrikannya. Pasangan yang terdiri dari dua kunci yang masing-masing berbeda, yakni privat dan public, membuat algoritma ini meskipun diberitahukan algoritmanya dan dipublikasikan kunci publiknya tetap akan terasa sulit untuk membongkarnya. Kecanggihan ini juga yang membuat algoritma ini dapat digunakan oleh public di internet dengan mudah dan transparan.

Dalam RSA ada yang disebut fase pembangkitan kunci, fase pengkodean, dan fase pendekodean.

Fase pembangkitan kunci

Pembangkitan kunci dilakukan dengan cara sebagai berikut:

1. Ambil bilangan p dan q yang merupakan bilangan prima (p dan q dirahasiakan)
2. n = bilangan p dikali q. (n tidak dirahasiakan)
3. ambil suatu bilangan disebut t, dimana t adalah bilangan yang dihasilkan dari perkalian (p-1) * (q-1)
4. Pilih kunci public, e yang relatif prima terhadap t.
5. Bangkitkan kunci privat dengan menggunakan persamaan
   e * d = 1 (mod (t))
   Kita harus mencari suatu nilai d yang memenuhi persamaan di atas.

Hasil dari algoritma di atas adalah :

1. Kunci public yang merupakan pasangan (e, n)
2. Kunci privat yang merupakan pasangan (d, n)

Fase Enkripsi dan Dekripsi

Enkripsi

1. Ambil kunci public penerima pesan, e, dan modulus n.
2. Nyatakan plainteks M menjadi blok-blok M1,M2, dst.
3. Setiap blok m dienkripsi menjadi blok cipher 'C' dengan rumus
   $C(i) = M(i)^e \bmod n$

Dekripsi

Setiap blok cipher C didekripsi menjadi blok M(i) dengan rumus sebagai berikut :

$M(i) = C(i)^d \bmod n$

Dari proses enkripsi dan dekripsi di atas terlihat bagaimana sebenarnya proses tersebut terjadi secara asimetrik dan dapat saling dipertukarkan.

Penggunaan RSA pada Sistem yang diajukan

Kunci public yang merupakan pasangan e,n adalah yang dipublikasikan kepada umum untuk digunakan sebagai pengkode pesan.

Kunci privat yang merupakan pasangan d,n adalah yang dirahasiakan. Kunci privat ini nantinya digunakan sebagai pendekode pesan (atau pembuka pesan)

Sebenarnya kedua kunci ini dapat saling dipertukarkan. Suatu pesan dapat dikunci dengan kunci public A, dan hanya dapat dibuka dengan kunci privat A, serta sebaliknya, suatu pesan yang dikunci dengan kunci privat A, hanya dapat dibuka dengan kunci public A. Aspek inilah yang membuat RSA menjadi sangat kaya dalam pemanfaatannya, salah satunya seperti yang dibahas dalam makalah ini.

Apabila suatu pesan dikunci dengan kunci public maka tujuannya jelas adalah untuk merahasiakan pesan. Namun jika dikunci dengan kunci privat maka tujuannya adalah untuk memberikan autentikasi atas pesan yang telah kita buat. Penggunaan keuntungan RSA ini secara bersamaan yakni merahasiakan pesan dan mengotentikasi pesan akan digunakan dalam rancangan utama system yang diajukan. Untuk lebih jelas dapat dilihat pada bagan di Lampiran Gambar 1.

Pada gambar terlihat bahwa ketika A ingin mengirim pesan kepada B, maka dia terlebih dahulu mengunci pesan dengan kunci public B (aspek confidential / kerahasiaan), kemudian hasil penguncian dengan kunci public B ini dikunci lagi dengan kunci privat A.

Di sisi penerima B, akan terjadi proses dekripsi sebagai berikut : (lihat Lampiran Gambar 2)

1. B akan membuka pesan yang terenkripsi dengan kunci publik A, dan akan mendapati pesan yang masih terenkripsi.
2. Pesan yang terenkripsi tersebut dari hasil 1 kemudian dibuka dengan kunci privat B.

Apabila hasil yang diperoleh dari langkah 2 adalah pesan yang benar atau setidaknya dapat terbaca dan dimengerti maka dapat dikatakan bahwa pesan tersebut (hasil dari langkah 1) adalah benar dikirim oleh A.

Dari diagram tersebut sekilas system memang tergolong sudah aman. Namun untuk memperkuat proses otentikasi dan non-repudiation dapat ditambahkan message digest, atau hash dengan SHA-1. Ini diperlukan untuk menjawab masalah yang terkait dengan otentikasi pesan, yakni mengotentikasi apakah pesan memang terubah atau tidak.

SHA-1

SHA adalah fungsi hast satu arah yang dibuat oleh NIST dan digunakan bersama DSS (Digital Signature Standard). Oleh NSA, SHA dinyatakan sebagai standar fungsi hash satu-arah. SHA didasarkan pada MD 4 yang dibuat oleh Rivest dari MIT.

Algoritma ini menerima masukan berupa pesan dengan ukuran maksimum 2^{64} bit dan menghasilkan digest yang panjangnya 160bit yang lebih panjang dari MD5.

Algoritma ini bekerja sebagai berikut :

1. Penambahan bit-bit pengganjal. Ini dilakukan agar panjang pesan kongruen dengan 448 modulo 512. Ini berarti panjang pesan setelah ditambahi bit penganjal adalah 64 bit kurang dari kelipatan 512.
2. Pesan dengan panjang 448 bit juga ditambah dengan bit-bit pengganjal. Ditambah dengan 512 bit pengganjal sebanyak 512 bit agar panjangnnya menjadi 960 bit.

3. Bit pengganjal adalah sebuah bit 1 diikuti selebihnya bit 0.
4. pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Panjang bit ini menggenapkan panjang pesan menjadi kelipatan 512bit.
5. SHA membutuhkan 5 buah penyangga uang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 5 x 32 bit = 160 bit.
6. Kelima penyangga tesebut diberi nama A,B,C,D,E dan diberi nilai inisiasi 8 hexa.
7. Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit.
8. Setiap blok 512bit ini diproses bersama dengan penyangga MD dari langkah 6 menjadi keluaran 128 bit.
9. Proses SHA-1 terdiri dari 80 putaran dan masing-masing putaran menggunakan bilangan penambah K_t yaitu :
   Putaran 0 – 19 mengunakan K_1
   Putaran 20-39 menggunakan K2
   Putaran 40-59 menggunakan K3
   Putaran 60-79 menggunakan K4
   K1- K4 adalah bilangan hexa yang dibangkitkan mirip dengan MD pada langkah 6.
10. Operasi dasar SHA ditulis dengan notasi sebagai berikut :
    A,B,C,D,E ← $(CLS_s(A) + Ft(B,C,D) + E + Wt + Kt)$, A, $CLS_{30}(B)$, C,D

    Dalam hal ini :

A,B,C,D,E	adalah lima buah penyangga 32 bit
t	adalah Putaran (0 – 79)
Ft	adalah fungsi logika tiap putaran yang berbeda-beda tiap 20 putaran
CLS_s	adalah operasi bit circular left sebanyak s bit
Wt	adalah word 32 bit yang diturunkan dari 512 blok bit yang sedang diproses
Kt	adalah konstanta penambah
+	operasi penjumlahan modulu 2^{32}.

    Untuk tiap putaran Ft adalah sebagai berikut :
    Putaran 0 -19 : (B and C) OR (~B and D)
    Putaran 20-39 : (B xor C xor D)
    Putaran 40-59 : (B and C) or (B and D) or (C and D)
    Putaran 60 – 79 : B xor C xor D

Hasil eksperimen yang telah dilakukan dengan menggunakan SHA-1 ini menunjukkan bahwa memang jika pesan diubah maka akan menghasilkan kode hash yang berbeda pula. (Eksperimen lengkap dapat dilihat pada dokumen Tugas Kecil 3 [MAN01])

Time Stamp

Time Stamp [RIZ01] adalah suatu metode untuk memperkuat penanganan penyangkalan. Timestamp ini berguna untuk menyatakan bahwa pesan benar telah dibuat pada waktu tertentu, tidak diubah dan tidak dikirim ulang. Untuk penggunaan timestamp ini sebenarnya telah ada protocol yang dibuat, sehingga memerlukan suatu pihak ketiga yang disebut Time Stamp Authority (selanjutnya disingkat dengan TSA) [PUR01].

Proses utama yang terjadi pada mekanisme timestamp ini adalah.

1. Pesan dihash dengan SHA-1.
2. Hasil digest 'M' ini dienkrip dengan menggunakan kunci public dari TSA.
3. Hasil langkah 2 ini selanjutnya dibuka dengan menggunakan kunci privat TSA
4. Hasil digest 'M' ini kemudian dikonkatenasi dengan waktu saat ini yang tercatat di TSA.
5. Setelah dikonkatenasi, maka hasil digest beserta hasil digest dari konkatenasi dikunci dengan kunci privat TSA.
6. TSA mengirimkan hasil konkatenasi beserta hasil langkah 5.

Apa yang menjadi kekuatan dari langkah pembuatan timestamp dengan TSA ini adalah :

1. Ada pihak ketiga yang menjadi penengah jika terjadi konflik antara pengirim dan penerima dalam hal penyangkalan
2. Yang diberikan kepada TSA adalah digest dari pesan sehingga TSA mungkin dapat melakukan pengubahan .
3. Hasil digest dikunci dengan menggunakan kunci public dan privat TSA adalah untuk meyakinkan bahwa memang hasil dari Timestamping itu adalah dari TSA. [STA01]

Kelemahan system ini adalah:
Bahwa TSA menjadi suatu objek yang mau tidak mau memang harus dipercayai, dan memang harus memiliki keamanan tinggi.

III. IMPLEMENTASI SISTEM

Setelah dipaparkan mengenai sendi-sendi dari elemen yang akan membangun system, maka selanjutnya adalah membuat suatu pengajuan implementasi. Pengajuan Implementasi dilakukan sebagai berikut :

Proses Enkripsi

1. Pesan asli sebelumnya dibuat digestnya kita sebut 'H'.

2. Selanjutnya H akan dikirim kepada TSA untuk diberi time stamp.
3. H juga dienkripsi dengan menggunakan kunci public dari Pengirim, untuk aspek otentikasi.
4. Di sisi TSA, dilakukan proses seperti yang tertulis di atas langkah-langkah memberikan timestamping TS.
5. Setelah timestamp diterima maka, hash dan time stamp disisipkan pada bagian pesan.
6. Pesan yang telah digabung dengan H + TS ini kemudian dienkripsi dengan kunci public penerima pesan.

Untuk lebih jelasnya, dapat melihat pada Gambar 3 bagian Lampiran.

Proses Dekripsi

1. Pesan yang terenkripsi E, pertama dibuka dengan menggunakan kunci privat penerima. Didapati pesan M, dan Hash H, serta timestamp TS.
2. Setelah pesan dibuka, maka langkah selanjutnya adalah melakukan otentikasi dengan mengecek hash dari pesan M dengan H, serta mengecek hasil timestamp dengan mengirim H + TS tersebut kepada TSA untuk dibandingkan (verifikasi).
3. Apabila ketiga hal ini dinyatakan terverifikasi dengan baik maka proses penanganan non repudiation dinyatakan berjalan dengan baik.

Untuk lebih jelasnya, dapat melihat pada Gambar 4 bagian Lampiran.

IV. PENUTUP

Analisis

Hasil analisis dengan eksperimen yang tertuang pada dokumen tugas besar 2 ([MAN02]) adalah menunjukkan bahwa memang pengubahan pada satu karakter pun pada pesan asli akan menyebabkan terubahnya hash, dan sekalipun hash dapat ditebak oleh penyerang, penyerang masih harus melalui proses pembongkaran kunci RSA.

Apa yang menjadi kekuatan pengajuan implementasi dengan system ini adalah gabungan kekuatan dari RSA itu sendiri ditambah dengan kekuatan SHA-1 dan penambahan kekuatan mediasi dari TSA.

Pada beberapa implementasi mengenai non repudiation, TSA ini sebagai pihak ketiga masih jarang diikutsertakan. Meski belum sempat dilakukan penghitungan seberapa sulit proses pembongkaran dengan menggunakan system yang diajukan ini, namun dengan menggunakan Triple RSA (pada pesan, hash, dan TS) serta ditambah dengan double SHA (pada pesan dan pada TS) maka secara sederhana dapat ditarik kesimpulan bahwa system ini akan sangat sulit untuk ditembus dan sulit untuk diserang dengan non repudiation attack.

Kesimpulan dan Saran

Pada makalah ini ditambahkan satu aspek baru yaitu TSA dalam proses penangangan non-repudiation sehingga memperkuat proses perlindungan terhadap kedua pihak yang sedang bertransaksi.

Proses penganganan non-repudiation dengan cara ini sebenarnya dapat dikembangkan lagi dalam penangangan perselisihan hak cipta, karena adanya timestamp.

Daftar REFERENSI

[MUN01] Munir, Rinaldi. Diktat Kuliah Kriptografi. Departemen Teknik Informatika Bandung 2005.

[MUN02] Munir, Rinaldi. Diktat Kuliah Struktur Diskrit. Departemen Teknik Informatika Bandung 2005.

[STA01] Stalling, William. Network Security. Mc Graw Hill. 2000.

[RIZ01] Riza, Lala Septem. Digital Timestamping: Suatu Tinjauan Komprehensif dan Usulan Model Skema Implementasi. ITB 2006.

[FAD01] Fadia, Ankit. Computer Security. Mc Graw Hill, 2000.

[PUR01] Purbo, Onno W. TCP/IP. Elex Media. 2000.

[MAN01] Manalu, Ecko dan Meliza Silalahi. Laporan Tugas Kecil 3 Kriptografi. ITB 2010.

[MAN02] Manalu, Ecko dan Meliza Silalahi. Laporan Tugas Besar 2 Kriptografi. ITB 2010.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

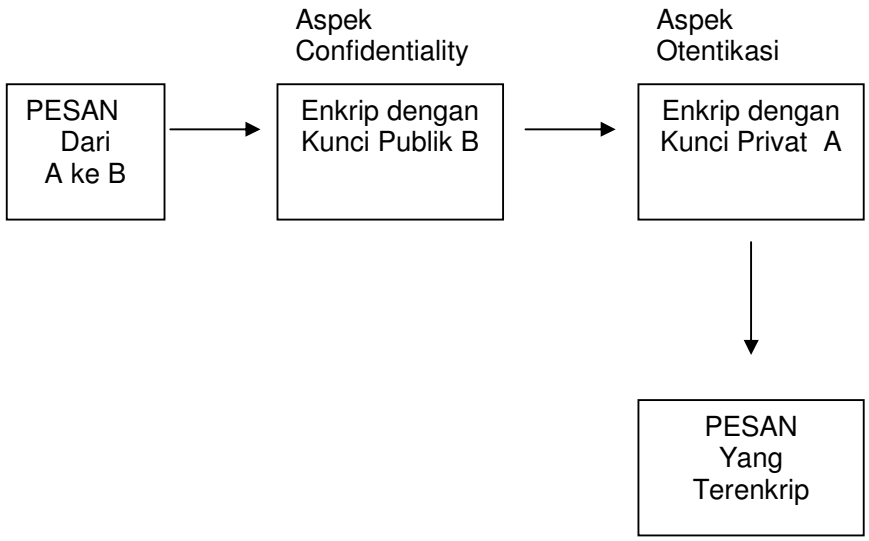
Bandung, 17 Mei 2010

Ttd

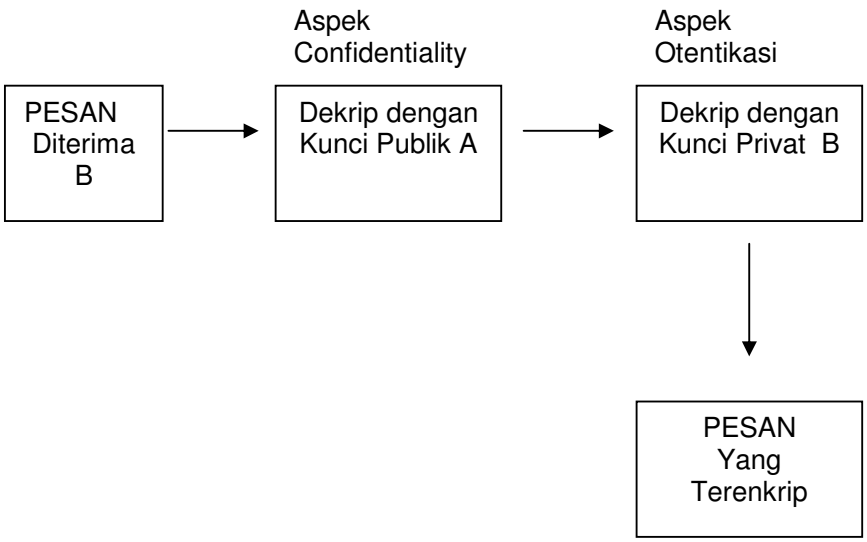
Ecko Fernando / 13508604

LAMPIRAN

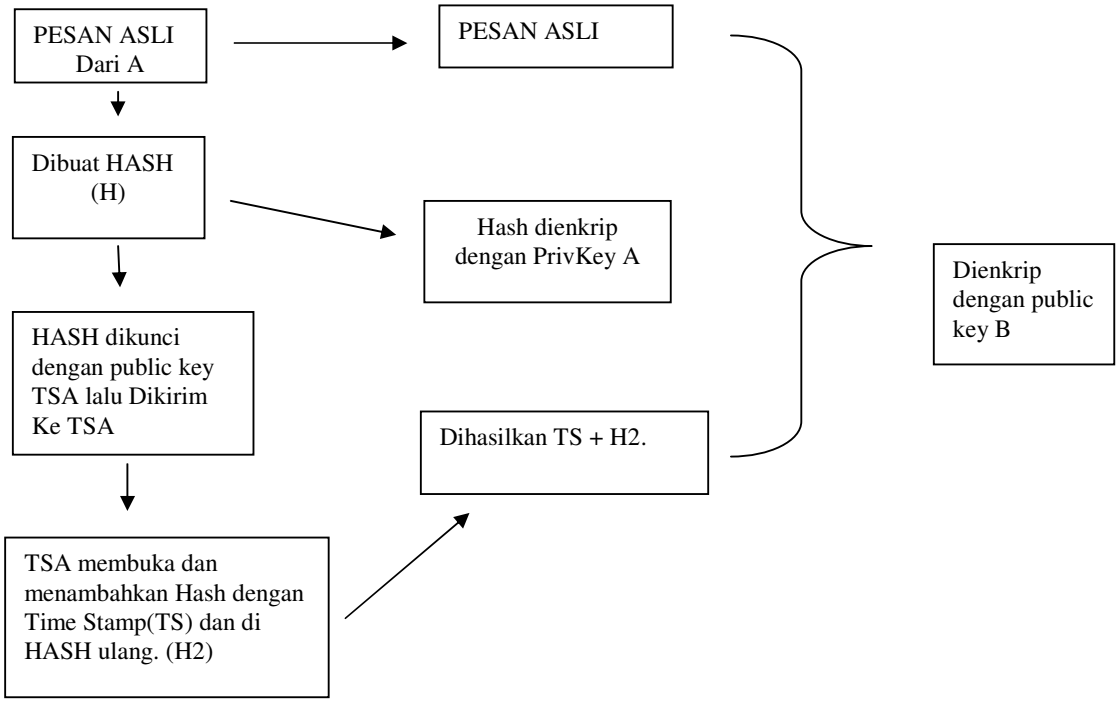
Gambar 1. RSA dengan Otentikasi dan Confidentiality



Gambar 2. RSA dengan Otentikasi dan Confidentiality



Gambar 3 Usulan Implementasi Sistem Enkripsi



Gambar 4 Gambaran Implementasi Sistem Dekripsi