Implementasi Algoritma RSA dan Three-Pass Protocol pada Sistem Pertukaran Pesan Rahasia

Aji Nugraha Santosa Kasmaji 13510092
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510092@std.stei.itb.ac.id

***Abstract*—Jaminan kemananan dalam proses pertukaran pesan adalah salah satu pemanfaatan dari kriptografi yang sering ditemui. Namun, untuk melakukan proses enkripsi dan dekripsi pada kedua pihak memerlukan kunci yang disepakati satu sama lainnya. Di dalam menyampaikan atau melakukan kunci tersebut, tidak dapat dijamin secara sepenuhnya bahwa kunci tersebut bersifat rahasia. Pada makalah ini akan ditawarkan sebuah kriptosistem yang memanfaatkan algoritma RSA dan *three pass protocol* dengan harapan mampu menyediakan sistem yang dapat melakukan pertukaran pesan yang aman tanpa perlu melakukan pertukaran kunci tertentu. Algoritma RSA digunakan untuk menjaga keamanan pesan, sedangkan *three pass protocol* sebagai aturan dalam melakukan pertukaran pesan. Modifikasi ringan ditambahkan untuk mengatasi masalah autentikasi, yaitu dengan menambahkan identitas berupa *username* dan *password*. Terakhir, dilakukan analisa keamanan lebih lanjut terhadap kriptosistem dengan membandingkan tingkat keamanan sistem terhadap sistem pertukaran data lain.**

***Index Terms*—kriptografi, kriptosistem, algoritma RSA, three pass protocol, pesan.**

I. PENDAHULUAN

Pertukaran informasi adalah sebuah proses yang tidak dapat dihindari oleh manusia, dalam memenuhi perannya sebagai seorang makhluk sosial. Pada era teknologi ini, proses pertukaran informasi dapat dilakukan dengan mudah, tidak terbatas pada lingkup waktu dan ruang karena informasi dapat dikirimkan lewat jalur digital. Namun, seiring dengan berkembangnya teknologi tersebut, muncul pula lubang-lubang pada yang dapat dieksploitasi lebih lanjut oleh pihak yang tidak bertanggung jawab. Karena proses pertukaran informasi tidak dilakukan secara fisik antara penerima dan pengirim pesan, maka dibutuhkan suatu sistem yang berfungsi untuk menjamin apakah pertukaran pesan tersebut aman, hanya dapat diterima oleh pihak yang berwenang saja.

Sampai saat ini, telah banyak metode-metode yang ditawarkan untuk menjaga keamanan dalam melakukan pertukaran informasi. Banyak jenis metode pada ilmu kriptografi yang menawarkan cara enkripsi dan dekripsi, dan aturan pertukaran pesan. Namun, kebanyakan proses yang ditawarkan sangat bergantung pada eksistensi data kunci yang disetujui oleh kedua belah pihak. Padahal, untuk dapat menerima data kunci tersebut, diperlukan juga proses pertukaran kunci yang aman. Kunci merupakan hal yang signifikan dalam proses enkripsi dekripsi, terutama untuk algoritma kriptografi kunci simetris.

Oleh karena itu, pada makalah ini akan ditawarkan sebuah kriptosistem yang menggabungkan beberapa metode untuk mencapai sebuah pertukaran informasi yang aman. Kriptosistem yang digunakan merupakan modifikasi dari aturan *three pass protocol*, khususnya *Shamir no-key protocol* dengan algoritma kriptografi RSA. *Shamir no-key protocol* memungkinkan terjadinya pertukaran pesan tanpa perlu melakukan perjanjian dan pertukaran kunci. Di sisi lain, RSA dengan harapan meningkatkan kompleksitas pengamanan pesan, sehingga pesan akan lebih susah dibongkar nantinya.

II. DASAR TEORI

A. Algoritma RSA

RSA adalah salah satu kriptosistem yang pertama kali digunakan untuk pertukaran data yang aman. Algoritma ini pertama kali ditemukan pada tahun 1977 oleh Ron RIvest, Adi Shamir, dan Leonard Adleman. Walau sudah berumur cukup tua, namun sampai saat ini RSA masih dipercaya memiliki tingkat keamanan yang dapat diandalkan. Pada kriptosistem yang ditawarkan RSA, kunci yang digunakan untuk melakukan dekripsi akan memiliki nilai yang berbeda dengan kunci untuk melakukan dekripsi. Kunci yang digunakan untuk melakukan enkripsi dapat disebarluaskan, sementara kunci dekripsi bersifat rahasia. Oleh karena hal tersebut, RSA dikatakan sebagai kriptosistem yang memiliki basis kunci publik, atau kunci nirsimetris.

Algoritma RSA menggunakan ilmu matematika dalam melakukan enkripsi dan dekripsi, dengan memanfaatkan kesulitan untuk melakukan faktorisasi dari hasil perkalian dua buah bilangan prima. Dua buah bilangan prima yang digunakan tentunya akan memiliki nilai yang cukup besar sebagai kunci publiknya, sehingga walau dengan menggunakan komputer sekalipun, akan dibutuhkan waktu dan resource yang cukup besar untuk melakukan penyerangan terhadap *ciphertext* yang dihasilkan.

Sebelum melakukan proses enkripsi dan dekripsi, algoritma RSA harus terlebih dahulu menyiapkan properti-properti awal melalui proses pembangkitan kunci. Tahapan dalam proses pembangkitan kunci dapat dilihat sebagai berikut:

1. Memilih dua buah bilangan prima yang berbeda, yang digambarkan dengan variabel *p* dan *q*, bersifat rahasia. Untuk alasan keamanan, diharapkan nilai *p* dan *q* memiliki panjang yang sama, dan dipilih secara acak.
2. Menghitung nilai *n*, yang merupakan hasil perkalian dari *p* dan *q*, tidak rahasia. Nilai *n* nantinya akan digunakan sebagai modulus baik untuk kunci public maupun kunci privat.
3. $\phi(n)$, nilai yang didapat dari perhitungan fungsi totient Euler ($\phi(n) = (p-1)(q-1)$), bersifat rahasia
4. Memilih sebuah angka integer *e* sebagai kunci publik, dengan syarat:
   - *e* memenuhi nilai $1 < e < \phi(n)$
   - *e* dan $\phi(n)$ harus relative prima, artinya factor pembagi terbesar antara *e* dan $\phi(n)$ adalah 1.
5. Mengkalkulasi nilai dari kunci privat *d* yang digunakan untuk melakukan dekripsi. Dengan syarat d adalah invers dari perkalian modulo dari *e* (mod $\phi(n)$), atau dapat dilihat sebagai berikut:
   - $d = e^{-1} \pmod{\phi(n)}$

   atau
   - $e * d = 1 \pmod{\phi(n)}$\

Setelah mendapatkan properti-properti yang diperlukan, maka kunci public akan disebarkan ke pihak publik. Proses enkripsi menggunakan publik akan melaui tahapan-tahapan berikut:

1. Nyatakan pesan menjadi blok-blok i, ada baiknya nilai *plaintext* dirubah menjadi bilangan integer: $m_1, m_2, m_3, \ldots$ (syarat: $0 < m_i < n-1$)
2. Hitung tiap blok *ciphertext* c_i untuk setiap blok *plaintext* p_i. Proses perhitungan dilakukan dengan persamaan:

$$c_i = m_i^e \textbf{ mod } n$$

3. Gabungkan tiap blok ciphertext kembali, dan pesan siap untuk dikirimkan

Sementara untuk melakukan dekripsi, digunakan kunci privat yang bersifat rahasia untuk satu orang tertentu. proses transformasi ke dalam blok yang digunakan sama dengan proses enkripsi. Sementara proses dekripsi memanfaatkan *Fermat's little theorem*, sehingga dapat dilakukan hanya dengan menggunakan persamaan berikut:

$$m_i = c_i^d \textbf{ mod } n$$

B. Three-Pass Protocol

Protokol adalah aturan yang berisi rangkaian langkah-langkah yang melibatkan dua atau lebih orang, yang dibuat untuk menyelesaikan suatu kegiatan. Dalam kriptografi, protokol digunakan oleh orang-orang yang terlibat untuk berbagai hal, seperti untuk proses otentifikasi, pengaktifan bilangan acak, bahkan untuk berbagi dan bertukar informasi rahasia.

Three-Pass Protocol adalah sebuah protokol kriptografi yang digunakan untuk mengirimkaan pesan rahasia. *Three-Pass Protocol* adalah sebuah kerangka kerja yang memungkinkan suatu pihak untuk mengirimkan pesan secara aman kepada pihak lainnya tanpa perlu melakukan pertukaran kunci. Disebut *three-pass* karena dalam prosesnya, pengirim dan penerima akan melakukan pertukaran tiga kali pesan yang dienkripsi.

Protokol ini menggunakan fungsi enkripsi *E* dan fungsi dekripsi *D*. fungsi enkripsi membutuhkan kunci *e* untuk merubah *plaintext m* menjadi sebuah pesan terenkripsi, (*ciphertext*) atau $E(e,m)$. Sementara fungsi dekripsi membutuhkan kunci *d*, yang digunakan untuk merubah *ciphertext* kembali menjadi *plaintext* semula, atau digambarkan dengan persamaan $D(d,E(e,m)) = m$.

Syarat utama sebuah fungsi enkripsi dan fungsi dekripsi dapat digunakan pada protokol ini adalah, persamaan fungsi enkripsi dan dekripsi dapat memenuhi sifat pada persamaan berikut:

$$D(d,E(k,E(e,m))) = E(k,m)$$

Dimana, *k* adalah kunci enkripsi yang bersifat independen terhadap fungsi enkripsi-dekripsi semula

Tahapan dalam melakukan pengiriman pesan menggunakan *Three-Pass Protocol* dapat dilihat sebagai berikut:

1. Pengirim pesan memilih kunci enkripsi privat *s* dan kunci dekripsinya yang bersesuaian, *t*. Pihak pengirim melakukan enkripsi terhadap pesan *m* menggunakan kunci *s* dan mengirimkan $E(s,m)$ kepada pihak penerima.
2. Penerima pesan memilih kunci enkripsi privat *r* dan kunci dekripsinya yang bersesuaian, *q*. dengan kunci enkripsi tersebut, pihak penerima melakukan *super-enrypt* terhadap pesan yang $E(s,m)$ yang diterimanya, lalu mengirimkan pesan yang telah dienkripsi ganda kembali ke pihak pengirim.
3. Pihak pengirim mendekripsi pesan balasan yang diterimanya dengan kunci dekripsinya. Karena sifat komutatif yang dimiliki Three-Pass Protocol, hasilnya adalah $E(r,m)$. Pihak pengirim mengirimkan kembali hasil dekripsi tersebut ke pihak penerima.

Pihak penerima nantinya akan melakukan dekripsi menggunakan kunci dekripsinya sendiri, sehingga didapatkan pesan awal *m* yang dikirimkan oleh pihak pengirim, tanpa perlu mengetahui atau bertukar kunci enkripsi dan dekripsi satu sama lainnya.

Aplikasi Three-Pass Protocol yang pertama ditemukan oleh Adi Shamir pada tahun 1980, dan disebut juga dengan *Shamir No-Key Protocol*. Algoritma yang digunakan Shamir juga memanfaatkan operasi pangkat dan modulo dari bilangan prima untuk fungsi enkripsi maupun fungsi dekripsinya.

Fungsi-fungsi yang digunakan dalam Shamir No-Key Protocol adalah sebagai berikut:

- Fungsi Enkripsi:

  $E(e,m) = m^e \bmod p$

- Fungsi Dekripsi:

  $D(d,m) = m^d \bmod p$, dengant $de \equiv 1 \pmod{p\text{-}1}$

Protokol buatan Shamir ini memiliki properti komutatif yang menjadi syarat dari *Three-Pass Protocol*, ang dapat dilihat dengan pembuktian sebagai berikut:

$$E(a,E(b,m)) = m^{ab} \bmod p = m^{ba} \bmod p = E(b,E(a,m)).$$

C. Two-Step Verification

Two-step verification atau disebut juga *two-factor authentication* adalah proses yang melibatkan dua tahapan untuk melakukan verifikasi identitas dari sebuah entitas yang ingin melakukan akses terhadap layanan yang diberikan oleh komputer atau pada sebuah jaringan tertentu. Sistem ini merupakan bentuk khusus dari *multi-factor authentication*, yang terdiri atas:

- Faktor pengetahuan
- Faktor kepemilikan
- Faktor inherensi (sifat/bagian konstan)

Dimana pada sistem ini, diambil dua dari tiga factor dengan tujuan untuk memperkuat sistem autentikasi agar lebih akurat dan tidak mudah ditembus.\

Contoh sederhana dari sistem ini adalah penggunaan *automated teller machine* (ATM), dimana dalam usaha untuk membuktikan identitas dari penggunanya, sistem membutuhkan dua buah hal, yaitu: ATM smartcard , yang merupakan factor kepemilikan, dan personal identification number (PIN), yang merupakan factor pengetahuan. Hal ini menyebabkan apabila pengguna kehilangan salah satu faktor saja, data miliknya masih cenderung aman dari potensi serangan yang ada.

III. DESKRIPSI SOLUSI

A. Kriteria Tujuan

Dalam menciptakan sebuah protokol kriptosistem ini, berikut adalah kriteria-kriteria tujuan yang diharapkan bisa dicapai:

1. Kerahasiaan (*confidentiality*)

   Layanan harus mampu menjaga isi pesan agar tidak dapat dibaca oleh siapapun yang tidak berhak untuk membacanya, namun tetap dapat dibaca oleh penerima yang diharapkan

2. Integritas data (*data integrity*)

   Layanan harus mampu mejamin apakah pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman. Dalam hal ini, diharapkan penerima pesan mampu menyadari apabila pesan mengalami modifikasi selama pengiriman.

3. Otentikasi (*authentication*)

   Layanan mampu mengidentifikasi kebenaran piha-pihak yang berkomunikasi. Apakah pengirim dan penerima pesan benar-benar orang yang diharapkan, dan bukan pihak ketiga yang memanipulasi pesan.

4. Nirpenyangkalan (*non-repudiation*)

   Layanan diharapkan mampu mencegah entitas yang berkomunikasi melakukan penyangkalan, baik dari pihak pengirim yang menyangkal bahwa ia telah mengirim pesan, maupun pihak penerima yang meyangkal bahwa ia telah menerima pesan.

Dalam proses perancangan protocol pertukaran pesan, diharapkan sistem mampu memenuhi keempat aspek tersebut sampai taraf ketahanan tertentu.

B. Rancangan Sistem Pertukaran Pesan

Pada bagian ini akan dibahas mengenai penyyusunan protocol yang ditawarkan dengan menggabungkan beberapa aturan dan algoritma tertentu untuk menambah kerumitan dan ketahanan sistem pertukaran pesan. Kontruksi yang digunakan merupakan gabungan antara *Shamir No-Key Protocol*, Algoritma RSA, dan beberapa modifikasi struktur sistem lainnya.

Proses pertukaran pesan secara umum akan menggunakan jenis *three-way protocol*, dimana akan terjadi tiga kali pertukaran data yang terenkripsi untuk setiap pengiriman pesan. Penggunaan kunci enkripsi dan kunci dekripsi pada setiap pihak nantinya diganti dengan menggunakan algoritma RSA, dengan harapan bahwa penghitungan faktorisasi terhadap hasil perkalian dua bilangan prima lebih sulit dilakukan dibandingkan penentuan suatu bilangan prima besar tertentu.

Untuk tambahan pada struktur sistem, ditambahkan penggunaan *username / ID* dan *password* untuk proses autentikasi tambahan, agar data yang dikirimkan benar-benar benar dapat diidentifikasi identitas pengirim dan penerimanya. Metode ini dilakukan untuk mencegah masalah otentikasi dan nirpenyangkalan dengan menggunakan pihak ketiga, yaitu pihak penyedia layanan.

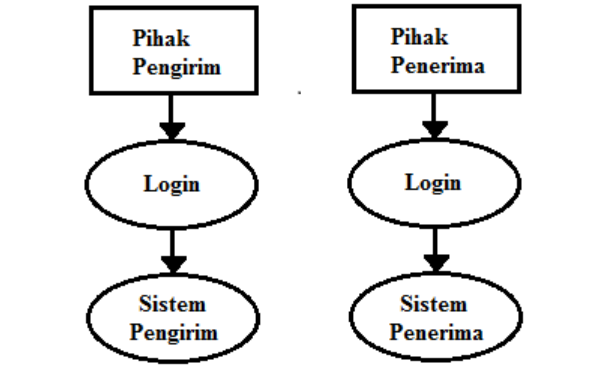
Untuk proses pengiriman pesan dapat lewat beberapa tahapan. Tahapan pertama, dimana pengirim akan mengirimkan pesan yang dienkripsi oleh aplikasi RSA miliknya. Tahapan kedua, dimana pihak penerima akan menerima pesan dan melakukan *super-encrypt* terhadap pesan tersebut (juga menggunakan algoritma RSA), dan akhirnya mengirimkan kembali ke pihak pengirim. Tahap ketiga, dimana pengirim akan melakukan dekripsi terhadap pesan balasan, dan mengirimkan kembali ke pihak penerima. Akhirnya, pihak penerima melakukan

dekripsi dan menerima pesan yang diharapkan.

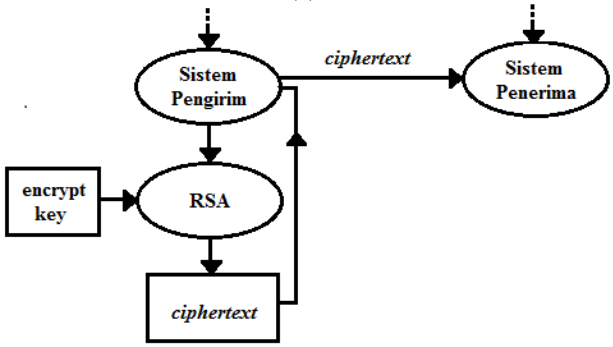
Dalam implementasinya, sistem diharuskan untuk memiliki beberapa fungsi utama, yaitu

- Fungsi untuk melakukan pembangkitan kunci yang digunakan untuk algoritma enkripsi dan dekripsi dengan metode RSA.
- Fungsi yang digunakan untuk melakukan enkripsi dengan metode RSA.
- Fungsi yang digunakan untuk melakukan dekripsi dengan metode RSA

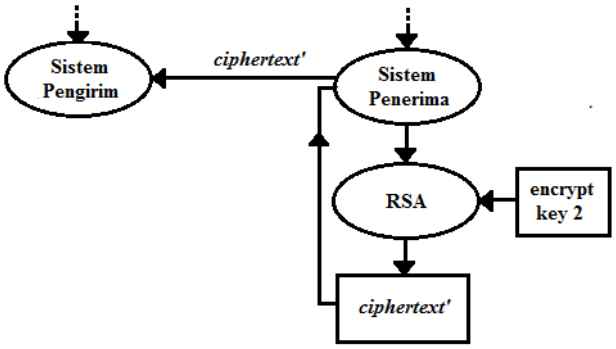
Selebihnya, adalah aturan atau tata cara bagaimana penggunaan fungsi-fungsi tersebut dapat membantu menambah pengamanan pertukaran pesan tanpa perlu melakukan pertukaran kunci.Untuk gambaran tata cara dan tahapan pertukaran pesan, dapat dilihat pada gambar di bawah ini:



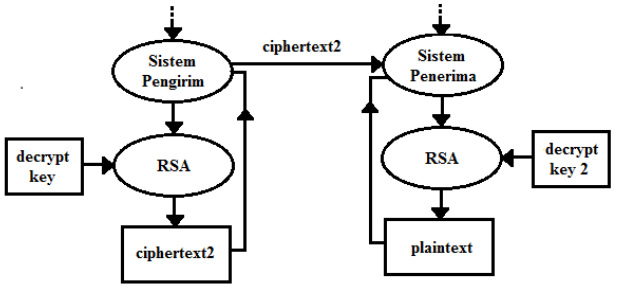
(a)



(b)



(c)



(d)

Gambar 1. Tahapan-tahapan protocol pertukaran pesan (a) otentikasi pengguna, baik pengirim dan penerima (b) proses pengiriman data pertama (b) proses pengiriman data kedua, pembalasan pesan (c) proses pengiriman data ketiga dan dekripsi pesan

Pada tahapan pertama, pengguna diharapkan untuk melakukan proses login untuk masuk ke dalam sistem. Proses login ini digunakan untuk menentukan apakah pihak pengirim dan penerima merupakan orang-orang yang berhak berhubungan dengan pesan. Proses ini juga digunakan untuk menempatkan nilai properti algoritma RSA pada setiap username, sehingga nilai kunci dapat bervariasi, sehingga harapannya akan sulit menerka banyak kombinasi kunci untuk keuda belah pihak.

Tahapan kedua adalah tahap pengiriman data pertama pada Three-Pass Protocol pada umumnya, dimana data akan dienkripsi dengan menggunakan kunci privat pengirim, lalu dikirimkan ke pihak penerima.

Tahapan ketiga adalah tahapan dimana pihak penerima melakukan *super-encrypt* terhadap pesan yang diterima dari pihak pengirim, lalu mengembalikannya pesan tersebut ke pihak pengirim.

Tahapan keempat, pihak pengirim melakukan dekripsi ulang terhadap pesan yang telah dienkripsi dua kali, lalu dikirimkan kembali ke pihak penerima. Dari situ, pihak penerima nantinya akan dapat melakukan dekripsi terakhir, yang menghasilkan pesan yang semula dikirimkan oleh pihak pengirim.

C. Analisis Awal

Dalam melakukan pengiriman pesan menggunakan *Three-Pass Protocol*, maka syarat utamanya adalah fungsi enkripsi dan fungsi dekripsi memiliki sifat yang komutatif. Sementara untuk RSA pada umumnya belum dapat dipastikan memiliki fungsi yang komutatif. Oleh karena itu, dibutuhkan modifikasi lebih lanjut, dimana memberikan dua buah kemungkinan agar RSA masih tetap dapat digunakan, yaitu:

- Melakukan proses enkripsi dan dekripsi dengan mendahulukan proses yang memiliki nilai modulo yang lebih kecil.
- Melakukan keseluruhan proses pengiriman pesan dengan menggunakan nilai modulo yang sama antara pihak pengirim dan pihak penerima.

Untuk percobaan ini, akan digunakan opsi yang kedua, artinya nilai modulo nantinya akan disediakan oleh pihak ketiga (sudah ditetapkan sebelumnya).

Sementara untuk faktor otentikasi dan nirpenyangkalan, penggunaan *two-factor authentication* digunakan untuk menambah ketahanan sistem. Faktor yang digunakan adalah faktor pengetahuan yang berupa *username* dan *password* masing-masing pengguna. Sementara faktor kedua adalah faktor inherensi, yang direpresentasikan dengan apa yang dimiliki oleh sistem. Harapannya dapat digunakan untuk nomor seri smartcard atau nomor barang khususnya untuk produksi sistem untuk telepon genggam. Namun, pada percobaan kali ini, nilai factor inherensi direpresentasikan dengan nomor seri sistem yang sebelumnya telah diberikan.

IV. IMPLEMENTASI DAN ANALISIS

A. Implementasi dan Pengujian

Dalam menghasilkan kunci, sistem melakukan generate untuk nilai modulus, serta kunci privat dan kunci publik dari masing-masing pihak terlebih dahulu. Dalam implementasinya, program yang dikembangkan menggunakan *library* milik BouncyCastle untuk menangani perhitungan integer dalam jumlah besar. Berikut adalah fungsi yang digunakan untuk melakukan pembentukan kunci dan nilai modulus:

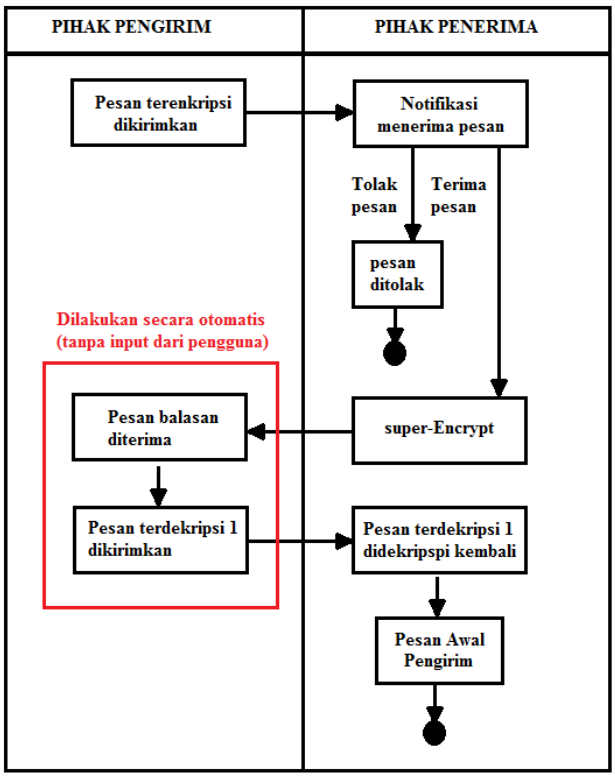
```
procedure generateKey(
    input
      BigInteger p,
      BigInteger q
    output
      BigInteger encrypt_key,
      BigInteger decrypt_key
)
KAMUS
    random_number = Random;
    one, totient = BigInteger;
    encrypt_key, decrypt_key = BigInteger;
ALGORITMA
    one  ← BigInteger("1")
    totient ← ( p – one ) * ( q – one )

    While (GCD(totient, encrypt_key)  =/= 1){
      encrypt_key  ← random_number()
    }

    decrypt_key ← modInverse(e,totient)
```

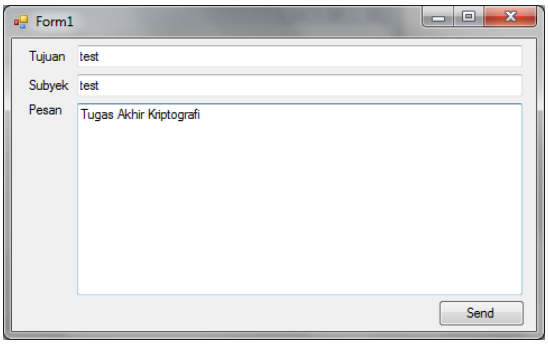
Untuk percobaan kali ini, digunakan beberapa nilai p, q, dan n yang degenerate di awal percobaan, namun dipakai secara konstan (belum otomatis melakukan pembentukan untuk setiap pengiriman).

Berikut adalah gambaran skema prosedur pertukaran yang dirancang untuk mengembangkan aplikasi:

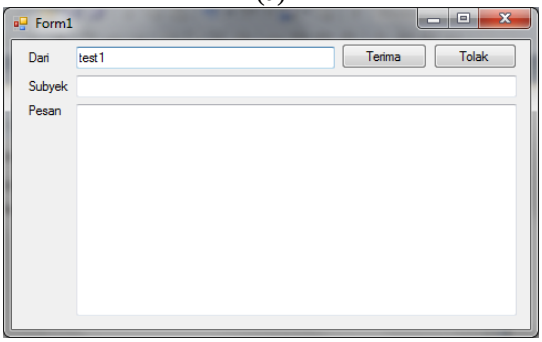


Gambar 2. Skema prosedur pertukaran data aplikasi

Batasan pengujian yang kedua adalah, aplikasi belum benar-benar tersambung melalui sebuah jaringan khusus, pengujian dilakukan secara *offline* dan manual. Sebagai pengganti jaringan, digunakan file eksternal sebagai sarana pertukaran data. Berikut adalah tampilan dari aplikasi yang digunakan selama pengujian:



(a)



(b)

Gambar 3. Tampilan aplikasi yang digunakan dalam pengujian (a) antarmuka aplikasi pesan (b) antarmuka aplikasi penerima pesan

B. Analisis Hasil Pengujian

Pada bagian ini akan dibahas hasil pengujian yang didapatkan per tahapan pengiriman pesan. Properti yang digunakan dalam pengujian dapat dilihat sebagai berikut:

p = 53

q = 41

n = 2173

e1 = 2049

d1 = 1409

e2 = 2057

d2 = 633

sementara untuk hasil enkripsi dan dekripsi pesan per tahapan proses pengiriman dapat dilihat sebagai berikut:

Tahap	Isi Pesan
Tahap Awal	Tugas Akhir Kriptografi
Nilai Byte	84 117 103 97 115 32 65 107 104 105 114 32 75 114 105 112 116 111 103 114 97 102 105
Tahap 1	2070 1580 1761 598 1840 1221 548 1909 1224 476 1836 1221 315 1836 476 1915 766 1670 1761 1836 598 1724 476
Tahap 2	238 2074 1399 937 1660 893 104 1644 1942 1589 1877 893 504 1877 1589 1032 627 669 1399 1877 937 487 1589
Tahap 3	2168 589 1935 596 1683 2000 445 1220 803 2013 442 2000 1118 442 2013 587 1159 458 1935 442 596 1632 2013
Byte Pesan Diterima	84 117 103 97 115 32 65 107 104 105 114 32 75 114 105 112 116 111 103 114 97 102 105
Pesan Akhir	Tugas Akhir Kriptografi

Berdasarkan data yang didapatkan di atas, dapat dilihat bahwa proses enkripsi dan dekripsi berjalan dengan cukup baik, sehingga dapat disimpulkan bahwa aspek kerahasiaan sudah cukup terjaga.

Namun, permasalahan yang cukup signifikan adalah letak titik lemah dari algoritma RSA komutatif, dimana bilangan modulo harus sama antara pihak pengirim dan pihak penerima. Hal ini mengakibatkan tingkat keamanan yang dihasilkan menurun secara cukup drastic. Apabila pihak pengirim dan penerima memiliki nilai modulo yang sama, maka pihak ketiga akan mampu mengidentifikasi nilai *d2* dalam waktu O(m), yang tidak memiliki tingkat keamanan lebih baik dari apa yang ditawarkan Shamir pada mulanya, yang bisa diserang lewat penghitungan dalam GF(p).

Titik lemah kedua adalah masalah autentikasi, walaupun sudah diperkuat dengan pengamanan dua tingkat, namun metode ini tidak dapat menanggulai gangguan dari *man-in-the-middle-attack*. Apabila pihak ketiga / penyadap mampu menyamarkan diri seolah-olah ia adalah pihak penerima, maka kecerobohan dari pihak pengirim dapat merusak keamanan pesan yang dikirimkan, walaupun dalam praktiknya proses ini memakan waktu cukup yang lama.

C. Hal yang Dapat Dikembangkan

Walaupun memiliki beberapa nilai lebih, seperti tidak perlu ada pertukaran kunci, namun beberapa hal dapat dikembangkan lebih jauh, meliputi:

1. Tingkat Keamanan Enkripsi, Fungsi Komutatif yang lebih baik
   Berdasarkan penelitian ini, algoritma RSA terbukti tidak memiliki tingkat keamanan yang cukup tinggi. Hal ini terbukti karena sifat komutatif pada RSA yang cukup dibatasi. Ke depannya, algoritma RSA dapat diganti dengan menggunakan algoritma Elgamal yang memiliki sifat komutatif yang lebih baik, sehingga aspek ketahanan yand didapatkan juga turut bertambah.

2. Penanganan Integritas Data
   Melihat seluruh struktur dan proses pertukaran data, aspek integritas data tidak terlalu ditonjolkan. Pesan yang telah berubah hanya dapat dideteksi apabila pada saat melakukan dekripsi, pesan yang didapat merupakan informasi yang rusak. Oleh karena itu, bisa ditambahkan proses pembentukan dan pengecekan *Message Authentication Code* (MAC), dengan metode yang cukup umum yaitu SHA-1.

V. KESIMPULAN

Pengujian terhadap sistem pengiriman pesan berbasis *Three-Pass Protocol* dan Algoritma RSA telah berhasil dan terbukti dapat dilakukan. Hanya saja masih memiliki beberapa kekurangan dan titik lemah yang bisa diperkuat lebih lanjut.

Untuk solusinya, salah satunya adalah mengganti algoritma enkripsi dekripsi dengan algoritma Elgamal yang memiliki sifat komutatif yang lebih baik, serta penambahan digest dengan metode SHA-1 untuk menjaga integritas data.

REFERENCES

[1] Rivest, R.; A. Shamir; L. Adleman (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" *Communications of the ACM* **21** (2): 120–126. doi:10.1145/359340.359342

[2] http://asecuritysite.com/encryption/comm. Waktu akses 19 Mei 2014 Pukul 21.33

[3] A. Menezes, P. VanOorschot, S. Vanstone (1996) *Handbook of Applied Cryptography* 500, 642.

[4] http://www.afn.org/~afn21533/keyexchg.htm. Waktu akses 19 Mei 2014 Pukul 21.33

[5] http://cryptocrats.com/crypto/new-algorithm-based-on-the-three-pass-protocol/. Waktu akses 19 Mei 2014 Pukul 21.33

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, 19 Mei 2014

Aji Nugraha Santosa Kasmaji (13510092)