Penerapan digital signature pada social media twitter

Arief Suharsono - 13510087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
arief.suharsono@comlabs.itb.ac.id

***Abstrak*—Pada zaman modern ini, social media sudah berkembang sangat pesat seiring dengan berkembangnya internet. Internet sudah menjadi jalur komunikasi yang menjadi kebutuhan hampir setiap orang. Namun internet yang sudah dijadikan jalur komunikasi universal ini masih memiliki kekurangan, yaitu penyebaran informasi oleh pihak-pihak yang tidak bertanggung jawab dengan menggunakan identitas orang lain, atau yang sering diserbut "pembajakan". Masalah ini dapat diselesaikan dengan menerapkan tanda tangan digital. Dalam pembuatan makalah ini, penulis mencoba melakukan implementasi tanda tangan digital pada jejaring sosial Twitter, dengan penerapan pada saat melakukan tweeting.**

***Kata Kunci*—Twitter, tanda tangan digital, tweeting.**

I. PENDAHULUAN

Pada zaman modern ini, social media telah berkembang pesat dan semakin banyak digunakan oleh orang-orang dari berbagai kalangan. Fenomena ini juga dimanfaatkan oleh para pemilik social media untuk menarik semakin banyak user, dengan cara menambahkan fitur fitur baru dan tingkat pengamanan jejaring social media yang bersangkutan. Perkembangan yang pesat dan semakin banyaknya pengguna ini menyebabkan social media sering digunakan sebagai alat untuk menyampaikan informasi secara massal. Social media juga sering digunakan untuk mempromosikan sesuatu yang baru kepada umum.

Salah satu media sosial yang sedang berkembang saat ini adalah twitter. Dimana twitter ini adalah social media dimana setiap user-nya dapat mengeluarkan tweet (dalam bahasa Indonesia artinya : kicauan), dan tweet tersebut akan muncul pada timeline user lain yang mem-follow user tersebut. Fungsi inilah yang menjadikan twitter marak digunakan untuk menyampaikan informasi secara massal, dimana ketika seorang user mengeluarkan tweet, maka akan muncul di timeline setiap user yang mem-follow user tersebut.

Namun, salah satu efek samping dari perkembangan yang pesat tersebut, twitter sering digunakan oleh orang-orang yang tidak bertanggungjawab untuk menyebarkan informasi palsu. Salah satu metodenya adalah mencuri akun orang tersebut lalu mengeluarkan tweet berisi informasi palsu seolah-olah pemilik user yang asli tersebut yang mengeluarkan. Istilah "pembajakan" juga semakin marak di kalangan anak muda, dimana akun twitter dari seorang user bisa diambil oleh orang lain dan mengeluarkan tweet yang aneh-aneh sehingga seolah-olah user tersebut yang mengeluarkan tweet tersebut.

Kasus di atas biasanya terjadi pada user yang melakukan login pada komputer yang bukan merupakan komputer pribadinya (misal : komputer milik temannya) dan lupa melakukan logout. Kemungkinan yang lain juga dapat terjadi jika user yang bersangkutan hanya melakukan login pada komputer pribadinya, namun lupa melakukan logout, dan komputer pribadinya tersebut diambil alih oleh orang lain (dipinjam, dicuri, atau digunakan tanpa izin).

Oleh karena itu, saya ingin mencoba melakukan pengamanan terhadap tweet pada twitter dengan menggunakan digital signature. Dimana saya akan membuat program yang bisa memberikan digital signature pada tweet yang dikeluarkan user. Digital signature tersebut akan dikirim beserta tweet yang diberikan. Tweet dari user akan muncul pada setiap timeline follower beserta dengan digital signature yang dibuat oleh user. Saya juga akan membuat program untuk mengecek digital signature dari setiap tweet yang ada timeline beserta kebenaran digital signature yang ditempelkan pada tweet tersebut.

Pembuatan digital signature dilakukan dengan membuat nilai hash dari message tweet yang akan dikirim. Nilai hash dibuat dengan algoritma SHA-1. Kemudian, nilai hash tersebut akan dienkripsi dengan algoritma ElGamal standar signature, dimana kunci publiknya akan disebarkan kepada umum dan kunci private nya akan disimpan sendiri. Nilai hash tersebut ditempelkan di akhir tweet, diawali dengan kode "<dss>" dan "<dsr>", dan diakhiri dengan kode "</dss>" dan "</dsr>".

Implementasi akan dilakukan dengan menggunakan membuat aplikasi dengan bahasa java. Dengan menggunakan algoritma-algoritma yang telah dibuat pada tugas besar II kuliah IF3058-Kriptografi. Untuk interkoneksi dengan server twitter, penulis mencoba menggunakan Twitter4J API, yaitu sebuah API untuk program berbasis java.

II. LANDASAN TEORI

A. Tanda Tangan Digital

Tanda tangan adalah sesuatu yang biasa digunakan pada dokumen dalam kehidupan sehari-hari. Hampir setiap dokumen (terutama dokumen resmi) terdapat tanda tangan (biasanya diletakkan di akhir dokumen). Kegunaan tanda tangan adalah untuk menunjukkan

bahwa pihak yang memberi tanda tangan sudah mengetahui dan menyetujui tentang adanya dokumen tersebut (menyetujui baik keberadaannya dan kebenaran isinya). Jika seseorang sudah memberikan tanda tangan pada sebuah dokumen, artinya orang tersebut ikut bertanggungjawab dengan keberadaan dokumen tersebut.

Konsep tanda tangan ini kemudian diterapkan pada dokumen-dokumen digital atau dokumen softcopy, yang akhirnya disebut sebagai "Tanda Tangan Digital" / "Digital Signature". Konsep tanda tangan digital ini digunakan untuk menandatangani dokumen-dokumen atau teks-teks yang bentuknya digital. Karena tanda tangan konvensional bersifat unik untuk setiap orang dan susah ditiru, maka konsep tanda tangan digital yang bagus juga harus unik dan susah ditiru oleh orang yang tidak berhak memberikan tanda tangan.

Konsep tanda tangan digital mengikuti konsep tanda tangan konvensional sebagai berikut :

- Tidak mudah dipalsukan dan bersifat otentik.
- Pemilik tanda tangan harus bertanggungjawab terhadap dokumen yang telah ditandatangani, dan tidak dapat menyangkal.
- Tanda tangan tidak mudah disalin / digunakan ulang untuk dokumen lain.
- Dokumen yang sudah diberi tanda tangan tidak dapat diubah.

Dalam makalah ini, penulis mencoba mengimplementasikan metode enkripsi tanda tangan digital menggunakan ElGamal digital signature scheme (seperti pada tugas besar II kuliah IF3058). Metode ini dipilih karena sudah pernah dibuat sebelumnya, dan dirasa masih cukup sulit untuk dipecahkan, walaupun sebenarnya metode ini masih jarang digunakan.

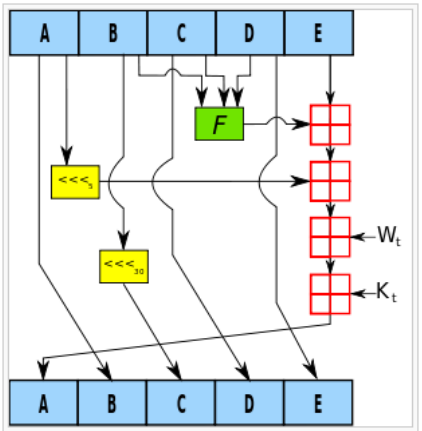
Hasil enkripsi dari metode ElGamal ini terdiri dari 2 bagian, yaitu bagian R, dan S. Pada implementasinya, kedua bagian ini akan dipisahkan.

B. Fungsi Hash SHA-1

SHA-1 (Secure Hash Algorithm-1) adalah fungsi kriptografik yang didesain oleh NSA dari Amerika Serikat. Fungsi SHA sendiri terdapat berbagai macam, mulai dari SHA-0, SHA-1, SHA-2, dan SHA-3. SHA-1 sendiri dianggap sebagai fungsi yang paling banyak digunakan dalam berbagai aplikasi jika dibandingkan dengan fungsi SHA yang lainnya.

SHA-1 membaca dokumen yang akan di-hash, dan menghasilkan 160-bit message digest (mirip seperti MD4 dan MD5). Message digest ini biasanya drepresentasikan ke dalam 40digit-heksadesimal (untuk mempersingkat message digest).

Algoritma SHA-1 secara ringkas adalah sebagai berikut :



One iteration within the SHA-1 compression function:
A, B, C, D and E are 32-bit words of the state;
F is a nonlinear function that varies;
$\lll_n$ denotes a left bit rotation by *n* places;
n varies for each operation;
W_t is the expanded message word of round t;
K_t is the round constant of round t;
$\boxplus$ denotes addition modulo 2^{32}.

Gambar 1: Skema Algoritma SHA-1

III. IMPLEMENTASI

A. Overview

Secara umum, pembuatan tanda tangan digital pada makalah ini dilakukan dengan menghitung nilai *hash* dari pesan yang akan dilindungi. Setelah itu, nilai *hash* tersebut dienkripsi dengan suatu algoritma ElGamal Digital Signature Scheme. Enkripsi menggunakan suatu kunci privat yang hanya diketahui oleh pembuat pesan. Hash yang telah terenkripsi inilah yang merupakan *digital signature* dari pesan.

Pada bagian *receiver*, penerima pesan akan memeriksa keaslian pesan dengan melakukan *hashing* pada pesan dengan algoritma *hash* yang sama. Setelah itu, *digital signature* akan didekripsikan dengan menggunakan kunci publik. Terakhir, kedua nilai *hash* yang didapatkan akan dibandingkan. Jika keduanya sama persis, berarti pesan yang diterima memang asli dari pengirim, atau tidak diubah-ubah oleh pihak ketiga. Sebaliknya, pesan telah diubah oleh pihak lain yang tidak bertanggung jawab.

Proses Enkripsi :

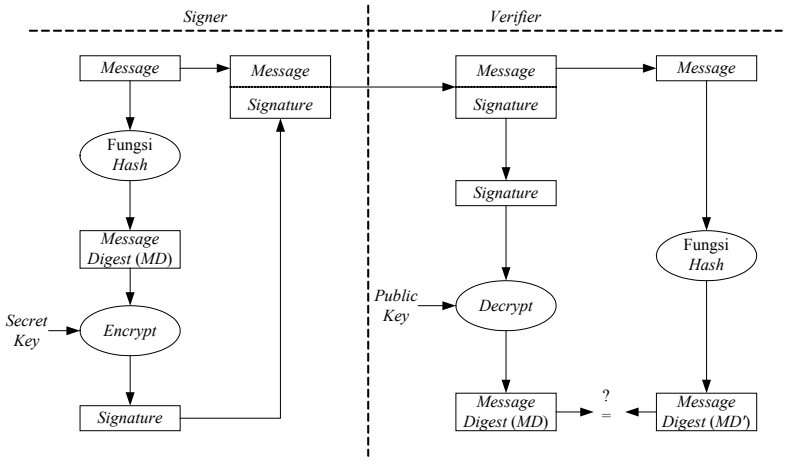
- Pengirim tweet meng-generate kunci publik dan kunci privat dengan algoritma ElGamal.
- User memasukkan pesan tweet yang akan dikirim.
- Pesan tweet akan dihitung nilai hash-nya, dengan menggunakan algoritma SHA-1.
- Nilai hash yang dihasilkan akan dienkripsi dengan menggunakan kunci privat.
- Hasil enkripsi akan ditempelkan pada bagian belakang pesan.

Pesan dikirim ke jejaring sosial Twitter.

Proses Dekripsi (Bila pesan tweet terenkripsi) :

- User mendapatkan tweet pada timeline twitter nya.

- User membuka kunci publik yang diberikan oleh pengirim tweet
- Pesan tweet akan dipisahkan terlebih dahulu, antara message dan signature nya.
- Message tweet dihitung nilai hash-nya dengan menggunakan metode SHA-1
- Digital signature didekripsi dengan menggunakan kunci publik dari pengirim tweet dan menghasilkan nilai hash dari message
- Nilai hash yang didapatkan dari dekripsi digital signature dibandingkan dengan nilai hash dari penghitungan nilai hash dari message.
- Jika perbandingan tersebut sama, maka message tersebut dapat disimpulkan autentik, jika tidak sama, maka message tersebut dapat disimpulkan tidak autentik.



Gambar 2 Skema Digital Signaturing

B. Strategi Penyelesaian Masalah

Salah satu permasalahan yang ada pada deskripsi adalah membuat suatu pembangkit kunci acak untuk algoritma ElGamal. Salah satu kunci yang digunakan haruslah bilangan prima. Oleh karena itu, terdapat sejumlah algoritma pembangkitan bilangan prima acak yang diimplementasikan pada program.

Pembangkitan bilangan prima acak menggunakan algoritma yang straightforward, yakni membangkitkan bilangan acak terlebih dahulu, lalu dilanjutkan dengan primality testing dengan Fermat's little theorem dan Miller-Rabin test.

Algoritma pada pembangkitan tandatangan digital dengan ElGamal juga memerlukan pencarian invers modulo. Penyelesaian dilakukan dengan menggunakan extended euclidean algorithm.

Aplikasi desktop dikembangkan dengan bahasa C#, sedangkan aplikasi email client yang digunakan untuk pengembangan add-in adalah Microsoft Outlook, yang juga dikembangkan dengan C#.

C. Struktur Data dan SubRutin

Karena dikembangkan dengan Java, maka seluruh program diimplementasi secara *object-oriented*. Pada kelas-kelas tingkat *model*, terdapat modul-modul yang terbagi berdasarkan prosedur pada program, yakni *key generator*, *signer* dan *verifier*.

Key generator berfungsi untuk membangkitkan key secara acak, *signer* digunakan untuk men-generate digital signature, dan *verifier* digunakan untuk memeriksa apakah pesan masih asli.

Spesifikasi masukan dan keluaran dari setiap modul:

- KeyGenerator
  - Input : N/A
  - Output : Kunci publik (p, g, y), Kunci privat (p, g, x)
- Signer
  - Input : Message, Kunci privat (p, g, x)
  - Output : Digital signature (r, s)
- Verifier
  - Input : Message, Digital signature (r, s), Kunci publik (p, g, y)
  - Output : Boolean, apakah message asli atau tidak

D. Contoh Kasus

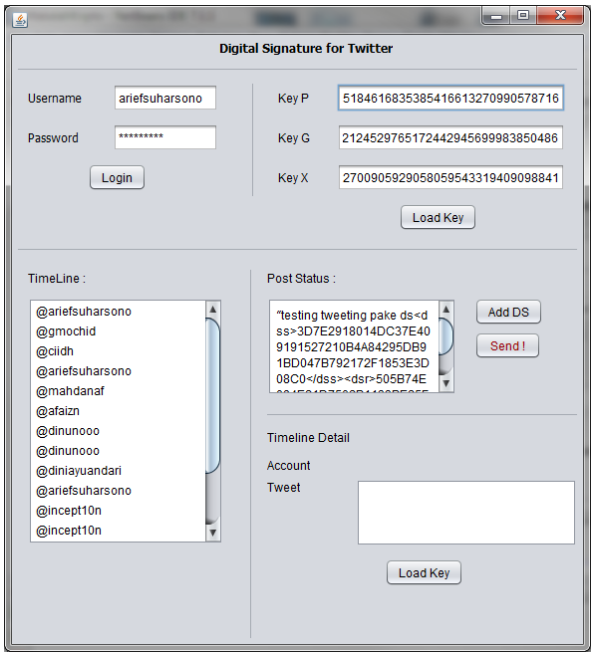
- Plainteks : "testing tweeting pake ds"
- Kunci Private P :
  5184616835385416613270990578716283082006619972085822983624665368714299088834 9
- Kunci Private G :
  2005949911537043910506043050209217591671843597212452976517244294569998385048 6
- Kunci Private X :
  4172350787864560228551937828098183084097862021543377468947824921215043566556 7
- Pemrosesan
  1. Penghitungan Nilai Hash :
     01CD097FEA27C3698E457EC8111920296F9618B
  2. Enkripsi :
     Hasil Digital Signature :

S =
3D7E2918014DC37E409191527210B4A84295
DB91BD047B792172F1853E3D08C0
R =
505B74E084E21D7502B1133BE25F27D6E5B
B23A371F6E714BD07F846D5C7E023

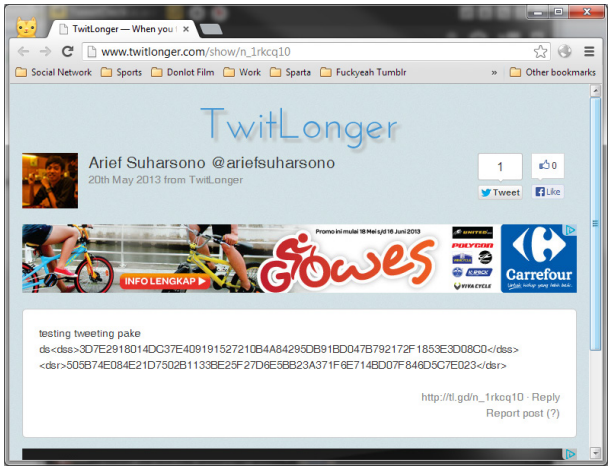
3. Penempelan Digital Signature
   Pesan yang dikirim :
   “testing tweeting pake
   ds<dss>3D7E2918014DC37E409191527210B4
   A84295DB91BD047B792172F1853E3D08C0</
   dss><dsr>505B74E084E21D7502B1133BE25F
   27D6E5BB23A371F6E714BD07F846D5C7E02
   3</dsr>”

IV. PENGUJIAN

Pengujian dilakukan dengan menggunakan kakas java buatan sendiri, dengan menggunakan Twitter4J API untuk koneksi dengan twitter melalui jaringan internet.

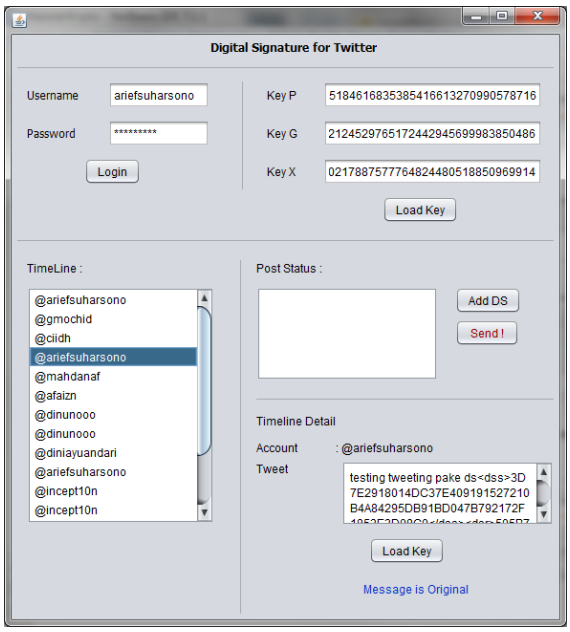


Gambar 3 Pengiriman Tweet



Gambar 4 Tweet yang sudah terkirim

Pengujian kedua, dilakukan dengan mencoba proses dekripsi, dan dapat dilihat bahwa kedua bagian cipherteks menghasilkan plainteks yang sama seperti semula.

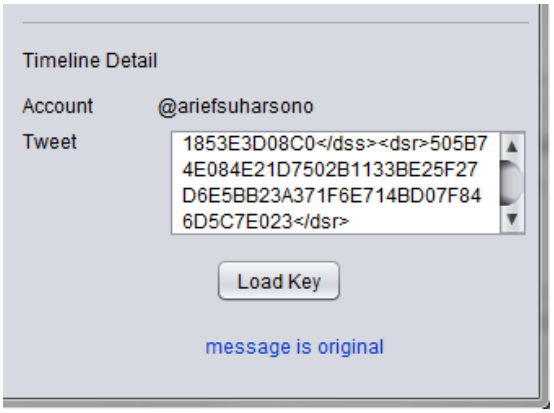


Gambar 5 Percobaan Verifying

V. ANALISIS HASIL

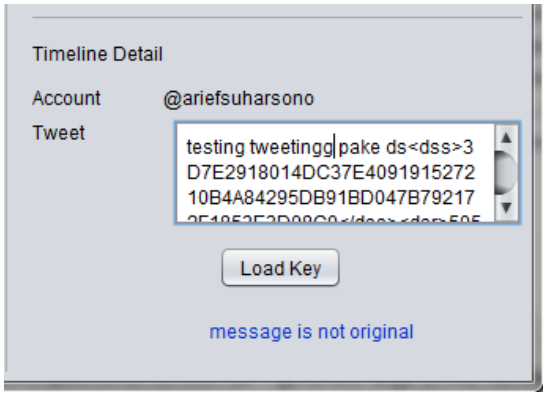
Pada bagian ini, penulis mencoba melakukan analisis hasil, untuk mengukur tingkat keamanan metode tanda tangan digital yang dibuat.

A. Original Message



Gambar 6 Tweet yang asli

B. Pengubahan 1 Karakter Pesan

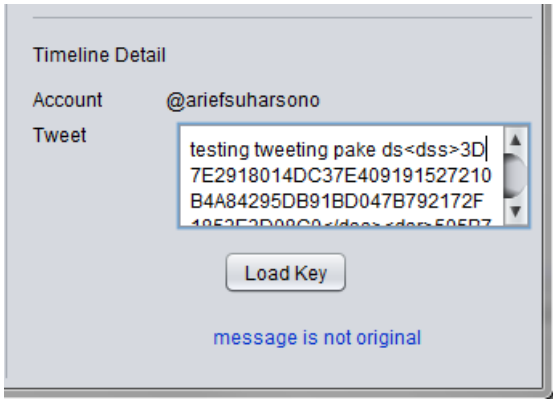


Gambar 7 Pengubahan 1 Karakter Pesan

C. Pengubahan 1 karakter Digital Signature

Gambar 8 Pengubahan 1 Karakter Digital Signature

D. Pengubahan 1 karakter kunci



Gambar 9 Pengubahan 1 Karakter Kunci

Dari percobaan tersebut, dapat disimpulkan bahwa metode yang dibuat sudah dapat memenuhi konsep tanda tangan digital yang baik, yaitu :

- Tidak mudah dipalsukan dan bersifat otentik.
- Pemilik tanda tangan harus bertanggungjawab terhadap dokumen yang telah ditandatangani, dan tidak dapat menyangkal.
- Tanda tangan tidak mudah disalin / digunakan ulang untuk dokumen lain.
- Dokumen yang sudah diberi tanda tangan tidak dapat diubah.

VI. KESIMPULAN

Digital Signaturing dengan metode SHA-1 dan ElGamal Signature Scheme dapat diaplikasikan dalam social media twitter. Meskipun dalam penerapannya akan membuat message jauh lebih panjang daripada message biasanya (panjang maksimal message twitter = 140 karakter), namun hal ini dapat diatasi dengan penggunaan third party application yang mendukung pengiriman tweet dengan jumlah karakter > 140.

Digital Signaturing dengan metode SHA-1 dan ElGamal Signature Scheme sudah termasuk metode digital signaturing yang baik, karena dapat mendeteksi perubahan pesan secara illegal, dan dapat mendeteksi pemalsuan tanda tangan.

VII. SARAN

Implementasi pada makalah ini dapat dikembangkan lebih lanjut untuk digunakan pada jejaring sosial lain ataupun platform aplikasi yang lain.

REFERENCES

[1] Munir, Rinaldi. 2011. "Bahan Kuliah IF3054 Kriptografi". Departemen Teknik Informatika, Institut Teknologi Bandung

[2] http://www.javacodegeeks.com/2012/03/twitter-api-on-your-java-application.html : Twitter API Tutorial
Waktu akses : 19 Mei 2013, 18:00 WIB

[3] http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/E94/182.PDF : ElGamal Signature Scheme.
waktu akses : 19 Mei 2013, 18:00 WIB

[4] http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf : Algoritma SHA-1
waktu akses : 19 Mei 2013, 18:00 WIB

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, 20 Mei 2013

Arief Suharsono - 13510087