ISSN 1693-1629
Edisi 6

Jurnal Ilmiah

ilmu komputer

Publikasi Hasil Penelitian diterbitkan oleh
Departemen Ilmu Komputer FMIPA, Institut Pertanian Bogor

1
Optimasi Jaringan Syaraf Tiruan dengan Algoritma Genetika untuk Peramalan Curah Hujan
Aziz Kustiyo, Agus Buono dan Novi Apriyanti

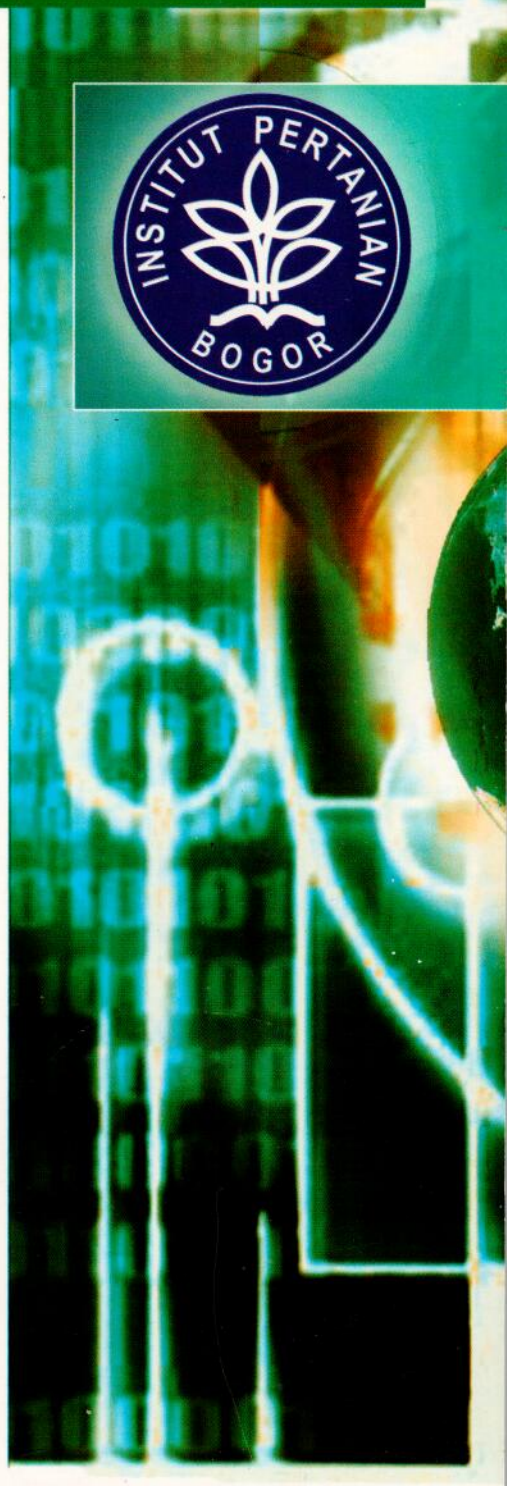
10
Pemodelan Simulasi HTTP Traffic Jaringan Komputer Lokal
F.X.Bayu Cahyo Raharjo, Panji Wasmana dan Fahren Bukhari

20
Pengkodean Aritmetika untuk Kompresi Data Teks
Solikha Nurhudayani, Fahren Bukhari dan Bib Paruhun Silalahi

29
Penggunaan Algoritma Challenge Response Identification untuk Autentikasi Entitas pada Intranet Messages
Uus Khusni, Wisnu Ananta Kusuma dan Sugi Guritman

37
Perancangan Engine Game 3 Dimensi dengan menggunakan Back Face, Viewing Frustum dan Occlusion Culling
Wikaria Gazali, Djunaidy Santoso dan Randy Tjandra

44
Penerapan Software Manufaktur untuk Peningkatan Produksi Hydraulic Excavator
Kudang B.Seminar, Endang N.Herdiana dan Widianingrum Mardirahayu



Vol.4 No.1 / Mei 2006

Jurnal Ilmiah **ilmu komputer**

Diterbitkan oleh: Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam - Institut Pertanian Bogor

Edisi 6 / Vol. 4. No. 1 . Mei 2006

ISSN : 1693-1629. Tanggal 4 April 2003

Susunan Redaksi

Sekretariat Jurnal Ilmiah **ilmu komputer :**

Departemen Ilmu Komputer FMIPA IPB
Jln. Raya Pajajaran, Kampus Baranangsiang Bogor 16144
Telp/Fax : 0251-356653 , E-mail : jurnal@ilkom.fmipa.ipb.ac.id
Rekening : Tabungan Taplus BNI Pajajaran Bogor.
No: **3031184** a.n.: *Annisa/Jurnal Ilkom*

Jurnal Ilmiah **ilmu komputer** *diterbitkan dua kali setahun, memuat tulisan ilmiah yang berhubungan dengan* ***bidang Ilmu Komputer****. Jurnal ini merupakan media publikasi ilmiah dan menerima tulisan dari luar IPB, berupa hasil penelitian atau bahasan tentang metodologi.*

Pihak perorangan / alumni yang telah memperoleh Jurnal Ilmu Komputer mohon mengganti biaya cetak Rp.50.000,-/expl, ditransfer melalui Tabungan Taplus BNI Pajajaran Bogor. No.Rek : 3031184 a.n.: Annisa / Jurnal Ilkom.

Daftar Isi

Sekapur Sirih i

Daftar Isi iii

Optimasi Jaringan Syaraf Tiruan dengan Algoritma Genetika untuk Peramalan Curah Hujan
Aziz Kustiyo, Agus Buono dan Novi Apriyanti 1

Pemodelan Simulasi HTTP *Traffic* Jaringan Komputer Lokal
F.X. Bayu Cahyo Raharjo, Panji Wasmana , dan Fahren Bukhari 10

Pengkodean Aritmetika untuk Kompresi Data Teks
Solikha Nurhudayani , Fahren Bukhari, dan Bib Paruhun Silalahi. 20

Penggunaan Algoritma *Challenge Response Identification* untuk Autentikasi Entitas pada *Intranet Messages*
Uus Khusni, Wisnu Ananta Kusuma dan Sugi Guritman 29

Perancangan *Engine Game* 3 Dimensi dengan menggunakan *Back Face, Viewing Frustum* dan *Occlusion Culling*
Wikaria Gazali, Djunaidy Santoso dan Randy Tjandra 37

Penerapan *Software* Manufaktur untuk Peningkatan Produksi *Hydraulic Excavator*
Kudang B. Seminar, Endang N. Herdiana dan Widianingrum Mardirahayu 44

Penggunaan Algoritma *Challenge Response Identification* untuk Autentikasi Entitas pada *Intranet Messages*

Uus Khusni[1], Wisnu Ananta Kusuma[2] dan Sugi Guritman[3]

[1]*Alumni Departemen Ilmu Komputer, FMIPA IPB*
[2]*Staf Pengajar Departemen Ilmu Komputer, FMIPA IPB*
[3]*Staf Pengajar Departemen Matematika, FMIPA IPB*

Abstrak

Password diasosiasikan dengan sebuah entitas, adapun pengertiannya adalah sebuah string yang biasanya terdiri dari 6-10 karakter yang mudah diingat oleh pemiliknya dan rahasia bersama dengan sistem, sebagai bukti keabsahan dari identitas seseorang. Tetapi orang kurang menyadari akan pentingnya suatu password yang aman agar password yang dimiliki tidak mudah diketahui. Oleh karena itu, dibutuhkan suatu mekanisme skema autentikasi yang kuat untuk melindungi password yang dimiliki pengguna.

Mekanisme Challenge Response dapat digunakan untuk menjaga kemananan password. Ide dari algoritma challenge response identification adalah pemilik pesan (claimant) dan penerima pesan (verifier) membuktikan keabsahan identitasnya masing-masing. Langkah ini dilakukan dengan mengadakan respon bagi suatu challenge tertentu. Challenge umumnya berupa karakter yang dipilih secara random dan rahasia. Karakter random yang digunakan berfungsi untuk menyediakan suatu keunikan dan jaminan untuk menghindari adanya serangan attacker.

Sistem yang dirancang adalah sejenis intranet messages yang berbasis client-server Sistem diimplementasikan dengan menggunakan bahasa pemprograman VB 6.0 dan microsoft acces xp sebagai DBMS. Server berfungsi sebagai penyimpan basis data yang berisi data mengenai identitas pengguna dan pesan yang dipertukarkan d iantara pengguna. Client adalah tempat pengguna meminta layanan fasilitas sistem. Fasilitas yang disediakan adalah fasilitas pengiriman pesan dan pengecekan pesan yang masuk untuk pengguna tersebut. Dalam sistem algoritma challenge response digunakan adalah algoritma challenge response dengan menggunakan kunci simetrik dan fungsi MAC sebagi autentikator. Tujuan yang ingin dicapai adalah entitas yang melakukan login ke sistem dijamin keabsahannya.

Kata kunci : *encryption, decryption, nessages authentication code, challenge response*

PENDAHULUAN

Latar Belakang

Kemajuan teknologi saat ini memudahkan orang untuk mengakses suatu sistem. Hal ini selain menguntungkan pengguna juga dapat beresiko tinggi bagi keamanan sistem. Jika sistem yang diakses berisi data yang penting dan rahasia, maka dibutuhkan suatu mekanisme pengamanan sistem untuk mencegah pengguna yang tidak memiliki hak akses (*intruder*).

Masalah tersebut dapat diatasi dengan adanya pembatasan hak akses menggunakan suatu skema yang disebut dengan skema autentikasi. Dalam skema autentikasi biasanya *password* digunakan sebagai bukti keabsahan seseorang untuk mengakses sumber daya sistem. Umumnya orang kurang menyadari akan pentingnya suatu *password* yang aman agar *password* yang dimiliki tidak mudah diketahui.

Password adalah alat autentifikasi yang lemah. *Intruder* biasanya dapat melakukan *brute force attack* untuk menemukan *password* tersebut. *Klein's Study* melaporkan bahwa pada tahun 1990, 24 % *password* dapat ditebak (Stanton 2006). Pada tahun 2000 persentase tersebut bertambah. *Cambride Study* melaporkan bahwa persentase *password* yang dapat ditebak mencapai 35 % (Stanton 2006).

Selain itu dalam suatu jaringan komputer, *password* sering tidak diproteksi. Protokol-protokol Internet seperti HTTP, FTP, Telnet, dan POP tidak memiliki mekanisme untuk memproteksi *password* (Stanton, 2006). Oleh karena itu dibutuhkan suatu mekanisme skema autentikasi yang kuat untuk melindungi *password* yang dimiliki pengguna.

Mekanisme *Challenge Response* dapat digunakan untuk memproteksi kemananan *password*. Ide dari *challenge response identification* adalah pemilik pesan (*claimant*) dan penerima pesan *(verifier)* membuktikan keabsahan identitasnya masing-masing. Langkah ini dilakukan dengan mengadakan respon bagi suatu *challenge* tertentu. *Challenge* umumnya suatu karakter yang dipilih secara acak dan rahasia. Karakter racak yang digunakan berfungsi untuk menyediakan suatu keunikan dan jaminan untuk menghindari adanya serangan *attacker*.

Pada penelitian ini mekanisme *challenge response* akan diimplementasikan untuk melakukan proteksi *password*. Prototipe sistem *intranet message* yang berbasis *client server* juga dibuat untuk membuktikan efektifitas *challenge response*.

Tujuan

Tujuan penelitian ini adalah:

1. mempelajari, memahami cara kerja, dan melakukan analisis algoritma *challenge response identification* berdasarkan kunci simetrik.

2. mempelajari dan memahami skema *password* secara umum, serta mekanisme tambahan yang bisa diimplementasikan di dalamnya.
3. mengimplementasikan algoritma *challenge response identification* pada suatu skema *password* sebagai alat untuk autentikasi entitas pada *intranet messages*.

Ruang lingkup

Penelitian ini meliputi analisis teori, analisis keamanan, dan analisis hasil implementasi algoritma *challenge response identification* dengan memakai kunci simetrik. Diasumsikan vektor awal (*Initial Vector* [IV]) sudah dijamin keamanannya.

TINJAUAN PUSTAKA

Kriptografi

Istilah kriptografi berasal dari bahasa Yunani yaitu *kryptos* yang artinya rahasia dan *graphein* yang artinya orang yang menulis. Schneier (1994) mendefinisikan kriptografi sebagai ilmu yang mempelajari teknik untuk menjaga keamanan pesan. Menurut Menezes *et al* (1996) kriptografi adalah studi matematika yang berkaitan dengan aspek keamanan informasi seperti kerahasian, integritas data, autentikasi entitas, dan autentikasi asal data. Sehingga terlihat bahwa kriptografi tidak hanya sebagai alat untuk keamanan informasi, melainkan juga berisi teknik atau prosedur yang berhubungan dengan keamanan informasi.

Kriptografi dapat memenuhi kebutuhan umum suatu transaksi yang meliputi:

1. kerahasiaan (*confidentiality*) yaitu menjaga kerahasian informasi dari siapapun kecuali yang berwenang mengetahuinya.
2. keutuhan (*integrity*) yaitu menjamin bahwa informasi tidak diubah oleh siapapun yang tidak berwenang.
3. transaksi yang tidak bisa disangkal (*non repudiation*) yaitu pencegahan dari pelanggaran kesepakatan-kesepakatan yang dibuat sebelumnya.
4. autentikasi (*authentication*) yaitu pelayanan yang berhubungan dengan identifikasi. Fungsi ini meliputi dua hal yaitu identifikasi entitas dan identifikasi pesan. Identifikasi entitas adalah adanya pembuktian yang kuat tentang identitas suatu entitas, sedangkan identifikasi pesan adalah pembuktian yang kuat bahwa pesan benar-benar berasal dari sumber informasi atau disebut autentikasi asal data.

Pesan asli yang belum disandikan disebut dengan *plaintext* disimbolkan dengan huruf *P* dan pesan yang sudah disandikan disebut *ciphertext* disimbolkan dengan *C*.

Enkripsi dan Dekripsi

Proses *kriptografi* dibagi menjadi dua yaitu proses enkripsi dan dekripsi (Menezes *et al* 1996). Enkripsi adalah proses untuk mengubah suatu *plaintext (P)* menjadi suatu *ciphertext (C)*. Notasi proses enkripsi adalah:

$$E(P) = C$$

Dekripsi adalah proses untuk mengubah suatu *ciphertext (C)* menjadi suatu *plaintext (P)*. Notasi proses dekripsi adalah:

$$D(C) = P$$

Proses enkripsi yang diikuti dengan proses dekripsi merupakan proses mengembalikan pesan *plaintext (P)* yang asli sehingga berlaku:

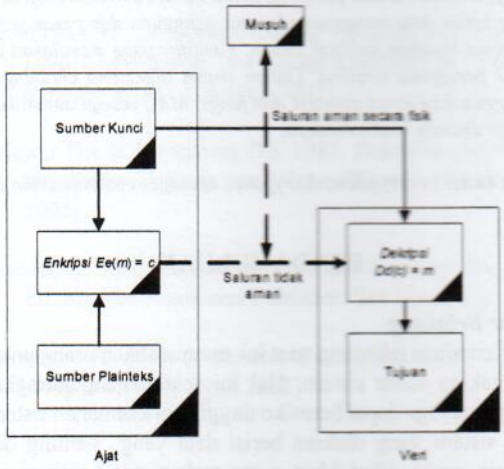
$$D(E(P)) = D(C) = P$$

Enkripsi Kunci Simetrik

Suatu skema algoritma enkripsi dikatakan algoritma simetrik, jika untuk setiap pasangan kunci enkripsi dan dekripsi *(e,d)*, maka secara komputasi *d* "mudah" dihitung apabila *e* diketahui, dan *e* "mudah" dihitung apabila *d* diketahui (Menezes *et al* 1996), namun pada prakteknya sering kali:

$$e = d$$

Sehingga algoritma simetrik disebut juga algoritma satu kunci. Gambar 1 menjelaskan algoritma enkripsi dengan memakai kunci simetrik.



Gambar 1. Enkripsi kunci simetrik (Guritman 2003).

Fungsi Hash

Secara umum fungsi *hash* diartikan sebagai fungsi yang memetakan *bitstring* dengan panjang acak ke *bitstring* dengan panjang tetap *(Guritman 2003)*. Namun berdasarkan atribut yang dimilikinya, fungsi *hash* dapat dibedakan menjadi beberapa definisi (Bakhtiari *et al* 1995). Definisi-definisi tersebut di antaranya adalah:

- fungsi *hash* satu arah (*One Way Hash Function* (OWHF))

  Fungsi *hash* satu arah adalah fungsi *hash* yang memetakan sebuah *string* dengan panjang acak kepanjang tetap. Atributnya adalah:

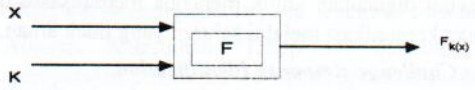
  a. dekripsi dari fungsi *hash* diketahui secara umum dan tidak membutuhkan informasi rahasia untuk pengoperasiannya.
  b. diberikan suatu fungsi *hash* dan pesan *M* maka mudah menghitung nilai *hash* dari pesan tersebut.

c. misalkan *h* adalah nilai dari suatu fungsi *hash* maka secara komputasi tidak mungkin menemukan *x* sehingga $H(x) = h$.

- fungsi *hash* bebas tumbukan (*Collision Free Hash Function* (CFHF))
  Fungsi *Hash* yang memetakan sebuah *string* dengan panjang acak ke panjang tetap disebut fungsi *hash* bebas tumbukan jika memenuhi atribut yang sama dengan fungsi *hash* satu arah dengan tambahan atribut sebagai berikut (Bakhtiar *et al* 1995):

  "Sangat sulit menemukan dua pesan yang berbeda *M* dan *M'* sehingga nilai *hash* dari kedua fungsi itu sama".

Fungsi *hash* dirancang sedemikian rupa sehingga sangat kecil kemungkinannya untuk mendapatkan nilai *hash* yang sama dari dua pesan yang berbeda. Oleh karena itu, penggunaan fungsi *hash* dalam bidang kriptografi biasanya dikaitkan dengan integritas data, perhitungan nilai *hash* dari suatu pesan dideskripsikan pada Gambar 2.



Gambar 2. Perhitungan nilai *hash* (Prasetya 2001).

$$H_i = F(X_i, h_{i-1})$$

dengan x = pesan yang dihitung *hash*nya
K = kunci untuk meng-*hash*
F = fungsi *hash*

Password dan Skema Password

Password diasosiasikan dengan sebuah entitas, adapun pengertiannya adalah sebuah *string* yang biasanya terdiri dari 6-10 karakter yang mudah diingat oleh pemiliknya dan rahasia bersama dengan sistem, sebagai bukti keabsahan dari identitas seseorang (Menezes *et al* 1996).

Skema *password* dapat dideskripsikan dengan pengguna memasukan pasangan ID-pengguna dan *password*, kemudian sistem akan membandingkan input yang dimasukan dengan data yang dimiliki untuk ID-pengguna yang dimaksud. Jika cocok maka *login* berhasil, jika tidak maka *login* ditolak. Dalam hal ini ID-pengguna merupakan klaim terhadap entitas dan *password* merupakan bukti pendukung klaim tersebut.

Ada beberapa teknik yang bisa digunakan dalam mendesain sebuah skema *password* yang baik (Menezes *et al* 1996):

1. Menyimpan password dalam basis data password
Teknik ini dikategorikan sebagai teknik bukan kriptografi dan merupakan suatu pendekatan yang sering digunakan. Teknik ini menyimpan *password* dalam bentuk teks asli (*cleartext*) disuatu basis data yang di *write* dan *read-protected (melalui pergantian atribut atau mode)*. Kelemahan teknik ini adalah sangat mudah diketahui oleh *user* yang memiliki hak akses penuh (*superuser*). Contohnya seorang admin yang berhasil mendapatkan hak sebagai *root*.

2. Mengenkripsi password
Teknik ini dilakukan dengan menyimpan *password* yang sudah ditransformasi menggunakan fungsi satu arah ke dalam basis data. Basis data untuk menyimpan *password* hanya perlu di *write-protected*.

3. Mengontrol pemilihan password yang akan digunakan
Teknik ini berguna untuk mencegah seorang pengguna menggunakan *password* yang "lemah" sehingga dapat diprediksi dengan serangan kamus (*dictionary attack*). Sistem perlu menerapkan aturan dalam pemilihan *password* sehingga aman untuk digunakan. Aturannya adalah sebagai berikut:

1. Jumlah karakter yang digunakan minimal 5 karakter.
2. Merupakan kombinasi dari beberapa huruf besar, huruf kecil, dan angka.
3. Bukan kata umum yang terdapat dalam kamus.
4. Bukan hal yang berhubungan dengan pengguna seperti nama pengguna, nomor telepon, dan lain-lain.
5. Mengganti *password* secara berkala *(misal 30 hari atau 90 hari sekali)*.

Serangan yang mungkin dilakukan terhadap teknik ini adalah pengunaan strategi serangan kamus yang sudah dimodifikasi untuk diterapkan pada pasangan *ID*-pengguna dan *password*-nya, sehingga bisa didapatkan suatu bentuk terlemah dari *password* yang masih memenuhi aturan yang ada.

4. Memperlambat pemetaan password
Teknik ini digunakan untuk memperlambat serangan pencarian lengkap *(Exhaustive Password Search Attack)* dengan membuat suatu fungsi verifikasi yang lebih sulit secara komputasi, misalnya memperbanyak jumlah iterasi yaitu output ke-i menjadi input iterasi ke-i+1. Jumlah iterasi tetap harus dibatasi sehingga tidak menghasilkan waktu tunda yang terlalu lama bagi pengguna yang sah.

5. Password salt
Teknik ini menggunakan penambahan *t-bitstring* acak yang sering disebut dengan *salt* pada *password*, sebelum dienkripsi. *Salt* dan *password* yang sudah di-*hash* disimpan dalam file *password*. Ketika pengguna menggunakan *password* untuk login ke sistem, sistem akan mencari *salt* dari pengguna tersebut, kemudian sistem mengenkripsi *password* dan *salt* menggunakan fungsi enkripsi tertentu dan mencocokan dengan data yang tersimpan. Cara ini tidak berpengaruh pada serangan pencarian lengkap karena *salt* disimpan dalam bentuk teks biasa dalam *file password*. Penambahan *salt* menambah kompleksitas serangan kamus dalam memprediksi *password*, yaitu secara simultan

menambah memori untuk menyimpan *password* tebakan sebesar 2^t variasi untuk masing-masing *password*, dan juga lama waktu yang diperlukan untuk memprediksi *password*.

6. ***Penggunaan frasa sebagai password***
   Untuk menghasilkan *password* yang lebih baik namun tidak menyulitkan pengguna dalam mengingat sebuah *password*, digunakan sebuah *frasa*. Pada teknik ini pengguna mengetikan sebuah kalimat atau *frasa* yang pendek untuk dijadikan sebagai *password*. Kekurangan dari teknik ini adalah bertambahnya jumlah karakter yang harus diketikkan oleh pengguna.

Messages Authentication Code (MAC)

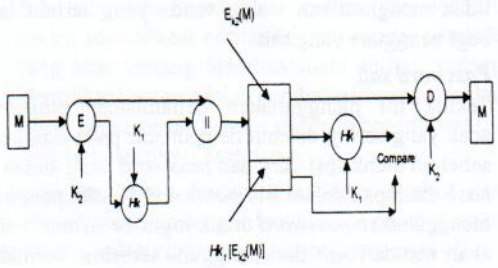
MAC adalah alternatif dalam teknik autentikasi dengan melibatkan kunci rahasia untuk menghasilkan suatu blok data dengan ukuran tertentu. MAC dikenal sebagai kriptografi *checksum*, yang ditambahkan pada pesan yang akan dikirm ke pengguna tujuan. Teknik ini diasumsikan bahwa dua orang yang bertukar pesan, sebut saja *A* dan *B* berbagi kunci rahasia *K*.

Contohn, *A* mengirim pesan pada *B* disertai dengan MAC. Setelah *B* menerima pesan dari *A*, *B* menghitung sendiri nilai MAC dengan memakai *input* pesan yang diterima dan kunci rahasia yang dibagi diantara mereka berdua. *B* membandingkan MAC yang diterima dari *A* dengan MAC yang dihitung sendiri. Jika hasil perbandingan MAC yang diterima sama dengan dikirim oleh *A*, maka pesan yang diterima tidak mengalami perubahan selama proses pengiriman. Jika hasil perbandingan MAC berbeda maka pesan telah mengalami perubahan.

Perhitungan nilai MAC dapat dinotasikan sebagai berikut:

M = Pesan yang akan di-*input*
Hk = Fungsi MAC
K = Kunci rahasia yang dibagi
MAC = *Messages Authentication Code*

Proses autentikasi dengan MAC dapat dilihat pada Gambar 3.



Gambar 3. Penggunaan MAC (Stalling 2003).

Kriptografi Protokol

Kriptografi protokol adalah suatu algoritma terdistribusi yang dapat digambarkan dengan serangkaian langkah–langkah yang harus dilakukan oleh dua entitas atau lebih, agar memperoleh suatu keamanan yang obyektif.

Misalnya, ada dua orang *Kurniawan* dan *Vieri* yang akan berkomunikasi. Mereka memilih suatu kunci simetrik yang akan digunakan dalam skema enkripsi untuk melakukan komunikasi melalui suatu saluran yang tidak aman. Langkah-langkahnya adalah (Menezes *et al* 1996):

1. *Kurniawan* membentuk suatu skema kriptografi kunci publik dan mengirimkan kunci publik ke *Vieri* melalui saluran tersebut.
2. *Vieri* membangkitkan suatu kunci yang akan digunakan dalam skema kriptografi kunci simetrik.
3. *Vieri* mengenkripsi kunci dengan menggunakan kunci publik milik *Kurniawan* dan mengirimkan hasil enkripsi kepada *Kurniawan*.
4. *Kurniawan* mendekripsi dengan menggunakan kunci pribadi dan memulihkan kunci simetrik.
5. *Kurniawan* dan *Vieri* siap berkomunikasi dengan privasi yang diperoleh dari sistem kunci simetrik dan kunci publik.

Dalam contoh diatas digunakan dua skema enkripsi yaitu enkripsi kunci simetrik dan enkripsi kunci publik. Dua kunci tersebut digunakan untuk mencoba merealisasikan keamananan komunikasi melalui saluran yang tidak aman.

Algoritma Challenge Response Identification

Ide dari algoritma *challenge response identification* adalah pemilik pesan (*Claimant*) dan penerima pesan (*Verifier*) membuktikan keabsahan identitasnya. Langkah ini dilakukan dengan mengadakan respon bagi suatu *challenge* tertentu. Respon tergantung pada kerahasian kedua entitas dan *challenge* yang diberikan. *Challenge* umumnya suatu karakter yang dipilih secara random dan rahasia. Karakter random yang digunakan berfungsi untuk menyediakan suatu keunikan dan jaminan untuk menghindari adanya serangan musuh.

Tujuan yang ingin dicapai yaitu bukti yang kuat bahwa pihak yang terlibat dalam komunikasi adalah pihak yang sebenarnya, sehingga protokol identifikasi diharapkan mampu berfungsi sebagai autentikasi entitas.

Mekanisme *challenge response* berdasarkan kunci simetrik antara pemilik pesan (*Claimant*) dan penerima pesan (*Verifier*) berbagi kunci simetrik agar tercapainya proses autentikasi. Misalkan, dua orang yaitu *Alice* dan *Bob* akan mengadakan suatu komunikasi. Keduanya telah berbagi kunci rahasia. Adapun langkah yang dilakukan untuk mengadakan proses autentikasi adalah (Menezes *et al* 1996):

$B \leftarrow A : ra$ (1)
$B \rightarrow A : rb, Hk(ra, rb, B^*)$ (2)
$B \leftarrow A : Hk(rb, A^*)$ (3)

A dan *B* adalah identitas dari entitas yang bersifat opsional yaitu bisa digunakan atau tidak. Jika akan digunakan maka *cleartext A* dan *B* harus disertakan. *A* disertakan dalam langkah 3 dan *B* pada langkah 2. Setelah langkah di atas dilakukan dengan benar, maka antara *Alice* dan *Bob* telah mengetahui identitas masing-masing dan siap berkomunikasi.

Brute Force Attack

Brute-force search atau *exhaustive key search* adalah suatu teknik dasar yang digunakan kriptoanalis untuk mencoba setiap kunci yang mungkin sampai ditemukan kunci yang sebenarnya. Hal yang membuat teknik ini sulit dilakukan adalah desainnya yang membuat seorang kriptoanalis harus memiliki mesin dengan kecepatan dan memori yang sangat besar. Jenis serangan ini sangat efisien untuk membongkar kunci rahasia pada kriptografi dengan ukuran blok kunci tetap.

Kriptoanalisis

Kriptoanalisis adalah suatu ilmu untuk mendapatkan *plaintext* dari pesan yang sudah dienkripsi tanpa memiliki kunci untuk memecahkan pesan tersebut. Orang yang melakukannya disebut kriptoanalis (Menezes *et al* 1996), sedangkan proses untuk melakukan kriptoanalisis disebut serangan (*attack*).

METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah:

1. Analisis Kebutuhan. Tahap ini dilakukan untuk mempelajari proses-proses atau fungsi utama dari sistem yang akan dibangun khususnya yang berkaitan dengan *Challenge Response Identification*
2. Perancangan. Tahap ini dilakukan untuk merancang arsitektur sistem, format input yang diperlukan, detil dari proses yang terdapat pada *Challenge Response Identification*, format *output* yang dihasilkan, dan basis data
3. Implementasi, yaitu tahap untuk mengkodekan hasil perancangan dengan bahasa pemrograman tertentu
4. Pengujian, yaitu tahap uji coba sistem untuk mengalisis kekurangan dan kelebihan sistem. Evaluasi sistem dilakukan setelah sistem selesai dibangun. Hal ini dilakukan untuk melihat apakah sistem tersebut telah memenuhi kebutuhan pengguna atau masih ada perbaikan.

HASIL DAN PEMBAHASAN

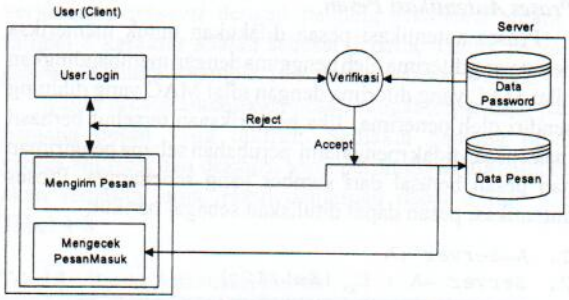
Fungsi Sistem

Sistem yang dikembangkan ini memiliki tiga fungsi utama, yaitu :

1. Autentikasi Entitas
2. Autentikasi Pesan
3. Pendistribusian *Session key*

Arsitektur Sistem

Sistem yang dirancang adalah sejenis *intranet messages* yang berbasis *client-server. Server* berfungsi sebagai basis data yang berisi data mengenai identitas pengguna dan pesan yang dipertukarkan di antara pengguna. *Client* adalah tempat pengguna meminta layanan fasilitas sistem, antara lain pengiriman pesan dan pemeriksaan pesan. Sistem yang dirancang dapat dilihat pada Gambar 4.



Gambar 4. Gambaran umum sistem

Perancangan input

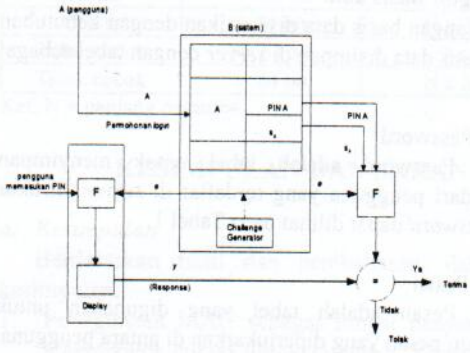
Sistem membutuhkan *input* dari pengguna yang akan digunakan dalam proses *autentikasi,* baik *autentikasi* pesan maupun *autentikasi entitas.* Proses *autentikasi entitas* membutuhkan *input password* dan kunci enkripsi dari pengguna. Sedangkan proses *autentikasi* pesan membutuhkan *input* kunci master dari pengguna. Pemasukan *input* dilakukan melalui *mouse* dan *keyboard,* dengan *user interface* berupa tampilan *windows.*

Perancangan Proses Autentikasi Entitas

Proses autentikasi entitas dilakukan pada saat pengguna melakukan *login* ke sistem. *Server* akan memeriksa *password* dan kunci enkripsi pengguna dengan data yang ada di *server.* Jika hasil pemeriksaan sama, maka *login* dari pengguna diterima. Langkah-langkahnya adalah:

1. *Client→Server*: *Username*
2. *Server→Client*: E_{kc} [*ke*||*r*]
   r = Bilangan Random
   ke= kunci untuk enkripsi dari server
3. *Client→ Server*:
   E_{ke} [*r*||PIN||*SA*||Autentikator]
   SA= *k* (PIN)
   Autentikator=*Hk*(*r*||Pin||*SA*||*IDc*)
   Hk=fungsi adalah MAC
4. *Server→Client*: Konfirmasi *login* diterima atau ditolak

Proses autentikasi entitas diperlihatkan Gambar 5.



Gambar 5. Proses autentikasi entitas.

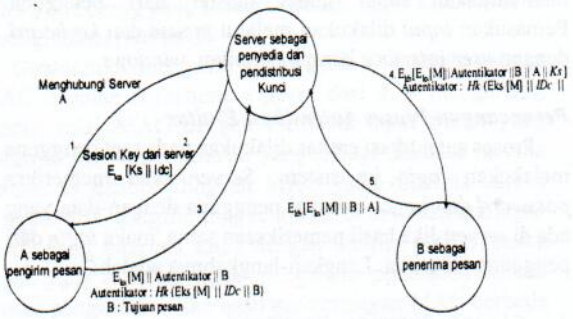
Proses Autentikasi Pesan

Proses autentikasi pesan dilakukan untuk memeriksa pesan yang diterima oleh pengguna dengan membandingkan nilai MAC yang diterima dengan nilai MAC yang dihitung sendiri oleh penerima. Jika pemeriksaan tersebut berhasil maka pesan tidak mengalami perubahan selama pengiriman dan pesan berasal dari sumber yang sebenarnya. Proses autentikasi pesan dapat dituliskan sebagai berikut:

1. A→*Server* :A
2. *Server* →A : E_{KA} [Ks||*IDc*]
3. A→*Server* : E_{Ks} [M]||Autentikator||B
   Autentikator = *Hk*(Eks[M]||*IDc*||*B*)
   B=Tujuan pesan
4. *Server*→:E_{Kb}[E_{Ks}[M]||Autentikator||B||A||*Ks*]
   Autentikator = *Hk*(Eks[M]||*IDc*||B)
5. B→*Server*: E_{Kb} [E_{Ks} [M]||B||A]

Perancangan Pendistribusian Session Key

Server berfungsi sebagai pendistribusi dan pembuat *session key*. Pendistribusianya dapat dilihat pada Gambar 6.



Gambar 6. Pendistribusian Kunci

Perancangan output

Tampilan *output* yang diberikan berupa tampilan hasil proses dari protokol yaitu tercapai atau tidaknya proses autentikasi entitas dari pengguna yang *login* ke sistem dan autentikasi pesan yang dipertukarkan oleh pengguna. Meskipun tidak secara detail menampilkan *output* untuk setiap langkah protokol, tetapi proses berjalannya protokol sampai tercapainya tujuan protokol dapat ditampilkan pada *client* berupa tampilan teks.

Perancangan Basis data

Perancangan basis data disesuaikan dengan kebutuhan sistem. Basis data disimpan di *server* dengan tabel sebagai berikut:

1. Tabel Password

Tabel Password adalah tabel untuk menyimpan identitas dari pengguna yang terdaftar di *server*. Strukur Tabel *Password* dapat dilihat pada Tabel 1.

2. Tabel Pesan

Tabel Pesan adalah tabel yang digunakan untuk menyimpan pesan yang dipertukarkan di antara pengguna. Tabel 2 menunjukkan struktur Tabel Pesan.

Tabel 1. Struktur Tabel Password

Nama Kolom	Tipe Data	Keterangan
Id_pwd	Auto Number	Nomor urut pengguna yang terdaftar di *server*
Nama	Text	Nama pengguna yang terdaftar di *server*
Pin_*user*	Text	*Password* yang dienkripsi. digunakan pada proses *autentikasi entitas*
Hash_pin	Text	Hasil enkripsi Pin *user* dengan memakai fungsi *MAC*
Kunci_master	Text	Kunci enkripsi untuk permohonan *session key* dari *server*

Tabel 2. Struktur Tabel Pesan

Nama Kolom	Tipe Data	Keterangan
Id_pesan	Auto Number	Nomor urut pesan yang ada di tabel pesan
Pengirim	Text	Nama pengguna yang mengirim pesan kepada pengguna yang lain
Penerima	Text	Nama pengguna yang menerima pesan pada pengguna yang lain
Subjek	Text	Judul dari pesan yang dikirmkan
Pesan	Test	Pesan yang dikirmkan kepada pengguna yang lain
Tanggal	Date	Waktu pengiriman pesan kepada pengguna lain

3. Tabel Koneksi

Tabel Koneksi adalah tabel untuk menyimpan *challenge* dan *session key* yang digunakan oleh *server* untuk autentikasi entitas pengguna yang *login* ke *server*. Struktur Tabel Koneksi diperlihatkan pada Tabel 3.

Tabel 3. Struktur Tabel Koneksi

Nama Kolom	Tipe Data	Keterangan
Id_koneksi	Auto Number	Nomor urut pengguna yang terhubung ke *server*
Nama	Text	Pengguna yang terhubung ke *server*
Challenge	Text	Bilangan random yang dibangkitkan oleh *server*
Alamat_ip	Text	Alamat IP pengguna yang terhubung ke *server*
Idc	Text	IDc dari pengguna ketika terhubung ke *server*
Kunci	Text	Kunci enkripsi untuk pertukaran pesan antara pengguna dan *server*

Implementasi Sistem

Hasil dari analisis masalah dan perancangan diaplikasikan pada suatu perangkat lunak dan perangkat penunjang lainnya. Adapun spesifikasi perangkat keras yang digunakan adalah:

- *Processor* Athlon 1,6 GHz
- RAM 384 MB
- Monitor dengan resolusi 600 x 800 pixels
- VGA *card* 64 MB
- *Harddisk* dengan kapasitas 40 GB

- *LAN Card On Board*
- *Mouse* dan *Keyboard*

Spesifikasi perangkat lunak yang digunakan adalah:

- Sistem Operasi Windows XP Professional
- Microsoft Access XP sebagai basis data
- Bahasa pemrograman Visual Basic 6.0

Pengujian Sistem

Pengujian ditujukan untuk menguji proses autentikasi entitas, autentikasi pesan dan waktu untuk autentikasi entitas serta autentikasi pesan.

Pengujian dilakukan dengan memakai dua buah komputer yang terhubung dalam suatu LAN. Satu komputer berfungsi sebagai *server* dan komputer lainnya sebagai *client*.

- ***Komputer 1***

Digunakan untuk menjalankan aplikasi *server* dengan spesifikasi sebagai berikut:

- *Processor* Athlon 1,6 GHz
- RAM 384 MB
- Monitor dengan resolusi 600 x 800 pixel
- VGA *card* 64 MB
- *Harddisk* dengan kapasitas 40 GB
- *LAN Card On Board*
- *Mouse* dan *Keyboard*
- Sistem operasi Windows XP Professional

- ***Komputer 2***

Digunakan untuk menjalankan aplikasi *client* dengan spesifikasi:

- *Processor* Pentium 4 1,7 GHz
- RAM 256 MB
- Monitor dengan resolusi 600 x 800 pixel
- *VGA card* 64 MB
- *Harddisk* dengan kapasitas 80 GB
- *LAN Card On Board*
- *Mouse* dan *Keyboard*
- Sistem operasi Windows XP Professional

Pada proses pengujian terlihat bahwa proses autentikasi entitas akan berhasil jika *password* dan kunci enkripsi yang dimasukkan oleh pengguna adalah benar dan cocok dengan yang tersimpan di *server*. Jika salah satu dari keduanya salah, maka *login* ditolak oleh *server*.

Proses autentikasi pesan berhasil jika kunci master yang dimasukkan oleh pengguna sama dengan kunci master yang tersimpan pada basis data di *server*. Jika kunci master tidak sama maka hasil dekripsi pesan akan berbeda sehingga perhitungan nilai *MAC* pesan akan berbeda.

Waktu proses autentikasi entitas tergantung pada kecocokan pasangan *password* dan kunci enkripsi yang dimasukan oleh pengguna dengan yang disimpan oleh *server* serta panjang *password* yang dimiliki oleh pengguna. Panjang *password* yang dimiliki pengguna dapat dibedakan menjadi:

X = Panjang *Password*

$1 \leq X \leq 8$ dan $9 < X \leq 16$

Waktu yang dibutuhkan oleh *server* untuk mengadakan verifikasi *password* dengan panjang *password* antara 1 sampai 8 karakter adalah sebesar 2 detik. Untuk panjang *password* antara 9 sampai 16 karakter sebesar 4 detik.

Waktu proses autentikasi pesan sangat tergantung pada panjang pesan yang dikirim. Semakin panjang pesan, proses enkripsi dan dekripsi serta perhitungan autentikator akan semakin lama. Hasil pengujian dapat dilihat pada Tabel 4-8

Tabel 4. Proses Autentikasi Entitas

Input Password dan Kunci Enkripsi	Verifikasi	*Login*
Kosong	Gagal	Ditolak
Ya	Berhasil	Diterima
Ya	Gagal	Ditolak

Tabel 5. Proses Autentikasi Pesan.

Input Kunci Master	Verifikasi	Autentikasi Pesan
Kosong	Tidak sama	Tidak sama
Ya	Sama	Sama
Ya	Tidak sama	Tidak sama

Tabel 6. Waktu Autentiksi Entitas untuk panjang *password* 1 sampai 8 karakter.

Input Password dan Kunci Enkripsi	Verifikasi	Waktu
Ya	Sama	2 s
Ya	Tidak Sama	N x 2 s
Kosong	Tidak sama	N x 2s

Tabel 7. Waktu Autentiksi Entitas untuk panjang *password* 9 sampai 16 karakter.

Input Password dan Kunci Enkripsi	Verifikasi	Waktu
Ya	Sama	4 s
Ya	Tidak Sama	N x 4 s
Kosong	Tidak sama	N x 4 s

Ket: N adalah banyak *login* yang dilakukan sampai berhasil.

Tabel 8 waktu autentikasi Pesan.

Kunci Master	Panjang Pesan	Waktu
Cocok	<= 64 bit	3 s
Cocok	> 64 bit	N x 3 s
Tidak cocok	<= 64 bit	3 s
Tidak cocok	> 64 bit	N x 3 s

Ket: N = panjang pesan/64

KESIMPULAN DAN SARAN

a. Kesimpulan

Berdasarkan hasil dan pembahasan, dapat ditarik kesimpulan:.

1. Penggunaan MAC sebagai fungsi penyandi untuk autentikator cukup untuk membuat skema *password* yang kuat, namun dengan mengimplementasikan

mekanisme lainnya lebih menjamin keamanan skema tersebut.

2. Setiap mekanisme memiliki manfaat masing-masing. Beberapa di antaranya:
   - Penyimpanan *password* dalam bentuk nilai *hash*-nya berfungsi untuk mempersulit *attacker* yang ingin mengetahui *password* pengguna.
   - Aturan pembuatan *password* berfungsi untuk memastikan bahwa *password* yang digunakan oleh pengguna merupakan *password* yang baik.
   - Penggunaan bilangan random dalam mekanisme autentikasi entitas berfungsi untuk memberikan keunikan dan *unpredictable*, sehingga mencegah serangan *attacker*.
4. Penggunaan mekanisme kunci simetrik pada algoritma *challenge response* memungkinkan adanya serangan *reflection attack*. Pada penelitian ini dicegah dengan penggunaan *session key* yang berbeda untuk setiap kali pertukaran pesan.

b. Saran

Pada penelitian selanjutnya diharapkan penggunaan algoritma enkripsi lainnya seperti algoritma Triple DES atau AES untuk dimplemetasikan pada mekanisme protokol *challenge response*.

DAFTAR PUSTAKA

Bakhtiar, S., Naini, R. S., Pieprzyk, J. 1995. *Cryptographic Hash Functions* : A Survey. citeseer.nj.nec.com/bakhtiari95cryptographic.html

Guritman, S. 2003. *Pengantar Kriptografi*. Jurusan Ilmu Komputer FMIPA IPB, Bogor.

Menezes, A. J., P. V. Oorschot and S. Vanstone. 1997. *Handbook of Applied Cryptography*. CRC Press Inc.

Prasetya, M. 2001. *Messages Digest 5 (MD5) dan Secure hash algorithm 1 (SHA1)* untuk *Autentikasi Pesan*. Jurusan Ilmu Komputer *FMIPA*.

Schneier, B. 1996. *Applied Cryptography Second Edition: Protocols, Algorithms and Source Code in C*. New York: Wiley.

Stallings, W. 2003. *Cryptography and Network Security Principles and Practice*. third edition., New Jersey: Pearson Education.

Stanton, J. 2006. *Network Security. Lecture Notes. Department of Computer Science, George Washington University*