

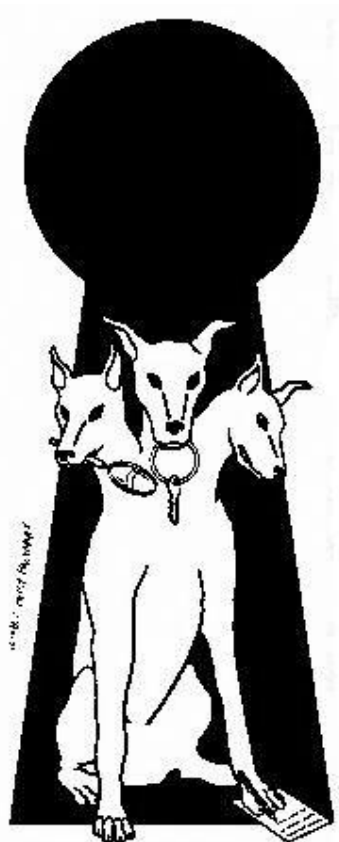
**LAPORAN PROYEK AKHIR  
EC 5010 KEAMANAN SISTEM INFORMASI  
SISTEM OTENTIKASI KERBEROS PADA JARINGAN  
KOMPUTER ITB**

**Dosen Pengajar: Budi Rahardjo**

**Oleh:  
Nama: Ivan Christian  
NIM: 13200160**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2004**



## Κερβερος

*... also spelled Cerberus. "n. The watch dog of Hades, whose duty it was to guard the entrance—against whom or what does not clearly appear; ... is known to have had three heads..."*

*—Ambrose Bierce, *The Enlarged Devil's Dictionary**

## ABSTRAK

Sebuah sistem jaringan terbuka tidak memiliki tingkat keamanan yang sama dengan sistem jaringan tertutup. Sistem otentikasi konvensional (*password-based*) rentan terhadap serangan seperti *eavesdropping*, *tampering*, dan *impersonation*. Untuk itu dibutuhkan sistem otentikasi yang lebih aman dan *scalable*.

Kerberos adalah protokol otentikasi jaringan yang dikembangkan oleh MIT. Protokol ini menggunakan kriptografi untuk otentikasi baik sisi *client* maupun *server*, sehingga diharapkan protokol ini mampu mengatasi kelemahan dari sistem otentikasi *password-based*.

Makalah ini akan membahas apa itu Kerberos, bagaimana cara protokol ini melakukan otentikasi pada jaringan, keunggulan dan keterbatasan dari protokol ini, dan struktur Kerberos pada jaringan yang besar. Dengan merujuk pada beberapa universitas yang telah mengimplementasikan Kerberos (seperti MIT dan Stanford), saya akan mencoba mendesain struktur sistem otentikasi Kerberos pada jaringan ITB.

# DAFTAR ISI

Abstrak.....	i
Daftar Isi.....	ii
Daftar Gambar.....	iv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang dan Rumusan Masalah.....	1
1.2 Tujuan Penulisan.....	3
1.3 Batasan Masalah.....	3
1.4 Sistematika Penulisan.....	3
BAB II SISTEM OTENTIKASI KERBEROS.....	5
2.1 Pendahuluan.....	5
2.2 Istilah Khusus dan Notasi dalam Kerberos.....	6
2.3 Asumsi-asumsi yang Berlaku.....	7
2.4 Komponen dari Kerberos.....	8
2.5 Cara Kerja Kerberos.....	10
2.5.1 Prinsip Dasar Kerberos.....	10
2.5.2 Format Ticket dan Fungsinya.....	11
2.5.3 Tahap-tahap Otentikasi.....	11
2.6 Keunggulan, Keterbatasan, dan Kelemahan Kerberos.....	16
2.6.1 Kunggulan.....	17
2.6.2 Keterbatasan.....	17
2.6.3 Kelemahan.....	18
BAB III OTENTIKASI ANTARJARINGAN KERBEROS.....	20

4.1 Pendahuluan.....	20
4.1 Aturan Penamaan Principal.....	20
4.1 Prinsip Kerja dari Cross-Realm Authentication.....	21
BAB IV USULAN STRUKTUR JARINGAN KERBEROS DI ITB.....	25
4.1 Pendahuluan.....	25
4.1 Kultur Jaringan Komputer di ITB.....	26
4.1 Usulan Struktur Jaringan Kerberos di ITB.....	28
4.1 Kendala yang akan Dihadapi.....	30
BAB V KESIMPULAN.....	32
5.1 Kesimpulan.....	32
REFERENSI.....	34

## DAFTAR GAMBAR

Gambar 2-1 Trusted Third-Party.....	10
Gambar 2-2. Tahap 1: Authentication Exchange.....	12
Gambar 2-3. Tahap 2: Ticket-granting Server (TGS) Exchange.....	13
Gambar 2-4. Tahap 3: Client/Server Exchange.....	14
Gambar 2-5. Tahap 4: Secure Communication.....	15
Gambar 2-6. Protokol Kerberos.....	16
Gambar 3-1. Model Cross-Realm Authentication pada Kerberos V4.....	22
Gambar 3-2. Model Cross-Realm Authentication pada Kerberos V5.....	23
Gambar 4-1. Peta Jaringan Komputer di ITB.....	27
Gambar 4-2. Usulan Struktur Jaringan Kerberos di ITB.....	29

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang dan Rumusan Masalah**

Kebutuhan untuk *sharing resource* telah mendorong terciptanya jaringan komputer. Keberadaan jaringan ini dapat meningkatkan kenyamanan dan efisiensi dari penggunaan *resource*. *Resource* yang dimaksud di sini dapat berupa *hardware* (CPU, *printer*, *memory*), program aplikasi (*e-mail*, *www*), bahkan informasi. Internet merupakan jaringan super yang melakukan *sharing resource* berupa informasi. Dalam skala yang lebih kecil kita mengenal jaringan lokal LAN yang melakukan *sharing resource* terbatas pada anggota kelompok tertentu saja.

Namun, meningkatnya kenyamanan akan menurunkan tingkat keamanan. Jaringan sistem terbuka (tidak terlindungi) sangat rentan terhadap serangan-serangan keamanan, seperti *eavesdropping*, *tampering*, dan *impersonation*. Untuk itu dibutuhkan sebuah sistem otentikasi jaringan yang handal. Tujuan dari sistem ini adalah untuk memastikan bahwa *user* yang meminta *service* adalah *user* yang sah. Selain itu otentikasi dapat digunakan untuk membatasi akses *user* pada jaringan.

Beberapa jenis sistem otentikasi telah dikembangkan. Salah satu diantaranya adalah *password-based authentication*. Pada sistem ini *user* diminta untuk memasukkan *username* dan *password* (yang seharusnya hanya diketahui oleh *user* yang sah saja), kemudian data tersebut dikirim melalui jaringan ke *server* yang bersangkutan. *Server* akan mengecek pada *database* kesamaan *username* dengan *password*. Jika benar, maka *user* tersebut memperoleh akses ke jaringan.

Namun, sistem otentikasi semacam ini tidak aman, karena beberapa orang dapat men-*tap* jaringan, menggunakan program *sniffer* untuk menangkap data, dan akhirnya diperoleh data *username* beserta *passwordnya* (*passive attack*). Akibatnya orang ini dapat mengaku sebagai *user* dan memperoleh akses seperti halnya *user* sebenarnya.

Untuk mengatasi kelemahan ini dibuatlah sistem *crypto-based authentication*. Di sini pengiriman data dilakukan dengan mengenkrip *username* dan *password* yang akan dikirim dengan kunci tertentu, dan kemudian didekrip di sisi *server*. Demikian pula halnya dengan data-data yang dikomunikasikan antara *user* dan *server*. Dengan begitu *passive attack* dapat diatasi karena yang disadap adalah *garbage* (karena penyerang tidak mengetahui kuncinya). Pada makalah ini akan dibahas salah satu protokol otentikasi jaringan yang disebut Kerberos. Kerberos merupakan sistem otentikasi jaringan berdasarkan kriptografi yang dikembangkan oleh Massachusetts Institute of Technology (MIT) pada Project Athena.

Struktur jaringan dengan sistem otentikasi Kerberos pada umumnya adalah terpusat. Bagaimana jika Kerberos ingin diimplementasikan di jaringan komputer ITB? Hal ini membutuhkan struktur yang tepat dikarenakan asal

mula jaringan komputer ITB yang bermunculan secara independen satu sama lain.

## 1.2 Tujuan Penulisan

Adapun tujuan dari penulisan makalah ini adalah:

- menjelaskan cara kerja sistem otentikasi Kerberos
- menjelaskan kelebihan dan kekurangan dari Kerberos
- memberi usulan mengenai rancangan struktur jaringan yang cocok untuk implementasi Kerberos pada jaringan komputer ITB.

## 1.3 Batasan Masalah

Dalam makalah ini masalah dibatasi hanya pada usulan mengenai struktur jaringan apabila sistem otentikasi Kerberos hendak diimplementasikan di jaringan komputer ITB.

## 1.4 Sistematika Penulisan

Bab I merupakan pendahuluan yang berisi tentang latar belakang, tujuan penulisan, batasan masalah, dan sistematika penulisan dari makalah ini.

Bab II menjelaskan mengenai sistem otentikasi Kerberos secara umum, yang meliputi cara kerja, asumsi yang berlaku, keunggulan dan kelemahan sistem Kerberos.

Bab III menjelaskan tentang *cross-realm authentication*, salah satu fitur Kerberos untuk melakukan otentikasi antar*realm*.

Bab IV membahas mengenai struktur jaringan komputer di tempat lain yang memiliki sistem otentikasi Kerberos, peta jaringan komputer ITB secara umum, dan bagaimana struktur jaringan yang sesuai dengan kultur jaringan ITB.

Bab V adalah kesimpulan dari pembahasan pada bab-bab sebelumnya.

## BAB II

### SISTEM OTENTIKASI KERBEROS

#### 2.1 Pendahuluan

Kerberos adalah protokol otentikasi yang menggunakan pihak ketiga yang dipercaya bersama-sama (*trusted third-party*). Protokol ini menawarkan otentikasi pada jaringan yang tidak aman. Kerberos Versi 1 sampai 3 digunakan secara internal dalam Project Athena, sedangkan Versi 4 diperuntukkan untuk digunakan secara umum. Oleh karena lingkungan yang berbeda dari Project Athena, maka dibuatlah Versi 5 yang merupakan perbaikan dari Versi 4 (tetapi tidak sepenuhnya *compatible*). Kini yang disebut standar protokol Kerberos adalah Kerberos Versi 5. Selain versi *freeware*nya (oleh MIT), Kerberos tersedia juga dalam bentuk versi komersial (oleh Microsoft, Sun, dll). Pada makalah ini dibahas Kerberos secara umum. Fitur-fitur khusus yang ada di Versi 4 dan 5 diluar ruang lingkup pembahasan.

Beberapa contoh service jaringan yang umumnya memerlukan otentikasi, antara lain:

- *printing*: umumnya penggunaan *printer* pada sebuah jaringan hanya diperuntukkan bagi anggota kelompok tertentu saja. Bahkan, *user* yang

melakukan *printing* dapat dikenakan biaya (misalkan pada *printer* di warnet) atau *user* dibatasi jumlah halaman yang mau dicetak.

- *remote file access*: hanya *user* sebenarnya yang berhak mengakses atau memodifikasi *file-file* yang tersimpan di jaringan.
- *remote login* (*rlogin*): hal ini berkenaan dengan akses *user* ke jaringan dari *workstation* di tempat lain.
- *window system*: untuk *user* tertentu tampilan *windows* dibatasi.
- *mail*: diinginkan hanya pemilik *e-mail address* saja yang dapat mengambil *e-mail* di POP3.
- *service management*

Perancangan Kerberos ditujukan untuk memberikan solusi bagi serangan-serangan keamanan yang tidak dapat diatasi oleh sistem otentikasi konvensional. Beberapa serangan yang ingin diatasi dengan perancangan Kerberos, antara lain:

1. *impersonation*, yaitu menggunakan *username* dan *password* yang bukan miliknya untuk memperoleh akses *service* dari jaringan.
2. *eavesdropping*, yaitu menyadap data-data yang lalu lalang di jaringan (*passive attack*).
3. *tampering*, yaitu mengambil data-data yang lalu lalang, mengubahnya, lalu mengirimkannya kembali (integritas data berubah).

## 2.2 Istilah Khusus dan Notasi dalam Kerberos

Kerberos menggunakan istilah-istilah khusus dalam dokumentasinya. Berikut ini adalah istilah-istilah penting yang umumnya digunakan, yaitu.

*authenticator*

ticket khusus yang dibuat oleh client untuk memperkuat otentikasi ticket yang dibuat oleh server Kerberos. Hanya dapat digunakan sekali saja, dan dienkripsi dengan session key.

*credential*

kumpulan ticket milik user yang digunakan untuk mendapatkan akses ke berbagai server di jaringan Kerberos. Credential ini dapat berupa ticket dan authenticator.

Kerberos Server

server khusus yang menjalankan fungsi otentikasi pada jaringan. Server ini terdiri atas tiga bagian, yaitu Authentication Server (AS), Ticket-granting Server (TGS), dan Key Distribution Center (KDC). Secara fisik, ketiganya dapat terletak di host yang sama atau berbeda.

*principal*

nama dari client dan server yang ikut ambil bagian dalam jaringan Kerberos.

*realm*

sebutan untuk jaringan yang menggunakan Kerberos.

*ticket*

identitas sementara yang dikeluarkan oleh server Kerberos. Ticket menjadi alat otentikasi antara client dan server tertentu. Berbeda dengan authenticator, ticket dapat digunakan berkali-kali sampai expired time habis dan dienkripsi dengan server key.

Masih banyak istilah lain yang dipakai sehubungan dengan Kerberos.

Namun, untuk penjelasan lebih detil silahkan merujuk pada [7].

### **2.3 Asumsi-asumsi yang Berlaku**

Dalam proses perancangannya Kerberos menggunakan asumsi-asumsi sebagai berikut:

1. *user* tidak menggunakan *password* yang mudah ditebak.

- Kata-kata yang terdapat dalam kamus, tanggal lahir, nama belakang, dan lain-lain sebaiknya tidak digunakan karena penyerang dapat melakukan *password guessing attack*.
2. jaringan komputer tidak aman, tetapi *workstation* “kurang lebih” aman.  
Maksudnya adalah *workstation* dan *server* tersebar di berbagai tempat sehingga pengamanan secara fisik pada jaringan tidak mungkin dilakukan. Posisi penyerang tidak berada diantara *user* dan *workstation*.
  3. otentikasi berlangsung dua arah (*mutual authentication*)  
Baik *client* maupun *server* membutuhkan jaminan identitas masing-masing.
  4. Kerberos Server berada di tempat yang aman secara fisik  
Diasumsikan bahwa hanya administrator yang mempunyai akses ke ruangan tempat Kerberos Server berada.

#### **2.4 Komponen dari Kerberos**

Secara umum program Kerberos memiliki beberapa komponen pembentuk, yaitu:

1. *Kerberos Application Library*  
Modul ini merupakan *library* yang menjadi antarmuka *client* dan *server*, antara lain berisi *routine* untuk membuat dan membaca permintaan otentikasi
2. *Encryption Library*  
Modul ini berisi *routine* untuk melakukan enkripsi dan dekripsi.
3. *Database Library*

Modul dengan *routine* untuk mengatur penyimpanan *database* Kerberos.

4. *Database Administration Program*

Program yang mengatur cara perubahan *database* Kerberos (penambahan dan penghapusan data-data *principal*).

5. *Administration Server*

*Server* untuk melayani perubahan isi *database* Kerberos. *Database* ini meliputi nama dan kunci privat dari *client* dan *server* yang memerlukan otentikasi melalui Kerberos.

6. *Authentication Server*

*Server* untuk melakukan otentikasi terhadap *principal-principal* dalam *database* dan membuat *session key*.

7. *Database Propagation Software*

Program untuk mengatur replikasi dari *database*. Tujuannya untuk meng-*update database* di setiap *server slave* berdasarkan *server master*.

8. *User Program*

Program di sisi *client* yang mengatur *user* untuk *login*, mengubah *password*, dan menampilkan atau menghancurkan *ticket* (biasanya dikenal dengan sebutan *kinit*).

9. *Application*

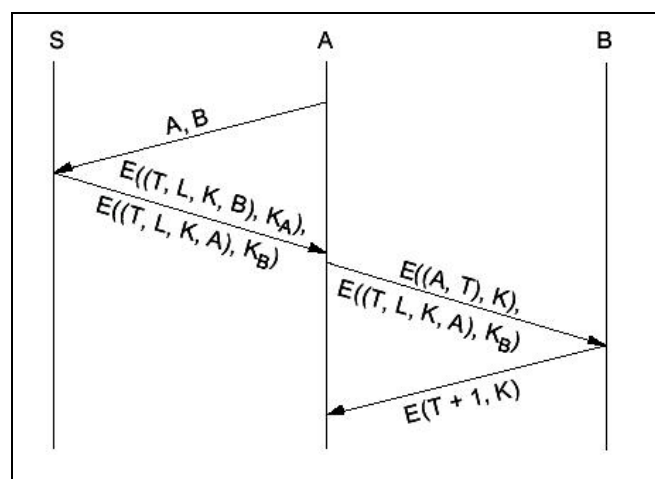
Program-program aplikasi yang memerlukan otentikasi untuk dapat mengaksesnya.

## 2.5 Cara Kerja Kerberos

### 2.5.1 Prinsip Dasar Kerberos

Kerberos bekerja berdasarkan model pendistribusian kunci yang dikembangkan oleh Needham dan Schroeder [3]. Kunci ini digunakan untuk mengenkrip dan mendekrip data yang akan dikirim melintasi jaringan. Jenis kriptografi yang digunakan adalah *symmetric key / secret key crypto* (versi asli dari Kerberos menggunakan algoritma *Data Encryption Standard (DES)*) [3]. Namun, dalam perkembangannya Kerberos juga dapat menggunakan *asymmetric key / public key crypto* [5].

Seperti yang telah disebutkan sebelumnya di subbab 2.1, Kerberos menggunakan prinsip *trusted third-party*. Artinya, baik *client* maupun *server* yang meminta otentikasi satu sama lain mempercayai apa yang dikatakan oleh Kerberos (pihak ketiga yang dipercaya). Hal ini diperlihatkan pada Gambar 2-1, dimana A dan B merupakan pihak yang hendak berkomunikasi. Otentikasi dilakukan melalui S yang memberikan bukti bahwa A yang hendak berkomunikasi adalah benar-benar A (bukan pihak lain).



Gambar 2-1 Trusted Third-Party

### 2.5.2 Format Ticket dan Fungsinya

Protokol Kerberos melakukan otentikasi dengan menggunakan *ticket* yang dikeluarkan oleh pihak ketiga dan *authenticator* yang dibuat sendiri oleh *client*. *Ticket* bersifat *reusable*, artinya dapat digunakan berulang-ulang sampai waktu kadaluarsanya habis. Secara umum *ticket* memiliki format dan fungsi sebagai berikut:

1. *username*

2. nama *server* yang dituju

Jika *server* yang dituju tidak mendapati namanya pada *ticket* setelah mendekrip *ticket* tersebut, maka *ticket* tidak valid.

3. *IP address workstation*

Keterangan *IP address* ini dapat digunakan untuk mencegah *impersonation* (misalkan *ticket* diperoleh dari hasil men-*tap* jaringan).

4. *time-stamp*

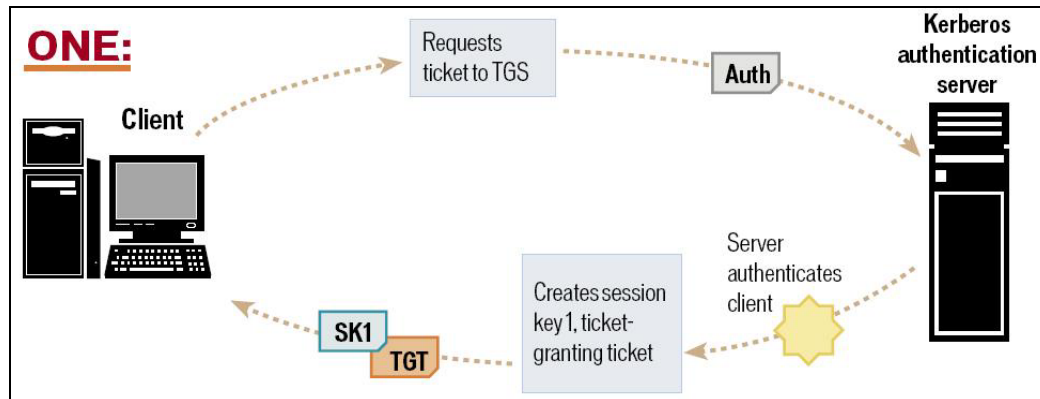
Digunakan bersamaan dengan *time-stamp* pada *Authenticator* untuk mencegah *user* lain melakukan *replay* terhadap *ticket*.

5. *expiration time*

Untuk mencegah *ticket* digunakan kembali jika *user* lupa menghancurkan *ticket*.

### 2.5.3 Tahap-tahap Otentikasi

Cara kerja Kerberos melakukan otentikasi dapat dibagi menjadi empat tahap. Secara garis besar tahap-tahap ini diperlihatkan pada Gambar 2-2 sampai Gambar 2-5 (sumber: [8]).

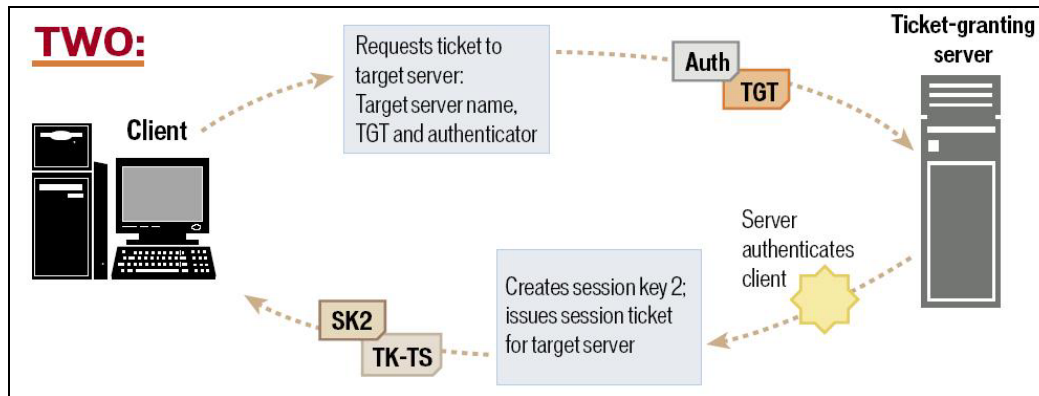


Gambar2-2. Tahap 1: Authentication Exchange

Tahap pertama disebut *Authentication Exchange*. Pihak yang terlibat adalah *client* dan *Kerberos Authentication Server (AS)*. Untuk *login* ke jaringan, program di sisi *client* (dikenal dengan *kinit*) akan meminta *user* untuk memasukkan *username* dan *password*. Program ini akan menurunkan *client key* ( $K_C$ ) dari *password* dan menghapus *password* sebenarnya di *workstation* tersebut. *Username* akan dikirim melintasi jaringan ke AS. Jika *username* terdapat di *database*, maka AS akan membuat *Session Key 1* (SK1 atau  $K_{C,TGS}$ ) untuk komunikasi antara *client* dan *Ticket-granting Server (TGS)*. Selain itu, AS juga membuat *ticket* untuk komunikasi antara *client* dan TGS (disebut *Ticket-granting Ticket* atau TGT atau  $T_{C,TGS}$ ). Selanjutnya  $K_{C,TGS}$  dan  $T_{C,TGS}$  dienkripsi dengan *TGS key* ( $K_{TGS}$ ). Paket ini diperuntukkan untuk dibuka hanya oleh TGS. Paket TGS dan  $K_{C,TGS}$  dienkripsi dengan  $K_C$ , lalu dikirimkan ke *client*. Notasi untuk proses ini dapat ditunjukkan seperti di bawah ini:

$$\{ K_{C,TGS}, \{ K_{C,TGS}, T_{C,TGS} \}_{K_{TGS}} \}_{K_C}$$

dimana  $\{ T_x \}_{K_x}$  berarti *ticket*  $T_x$  dienkripsi dengan kunci  $K_x$ . Warna merah menunjukkan paket TGS yang dienkripsi dengan  $K_{TGS}$ .



Gambar 2-3. Tahap 2: Ticket-granting Server (TGS) Exchange

Tahap berikutnya disebut *TGS Exchange*. Data dari AS didekripsi dengan menggunakan  $K_C$ . Jika *password* yang dimasukkan sesuai dengan *username*, maka *client* akan mampu mendekripsi data dengan benar. *Client* akan mendapatkan  $K_{C,TGS}$  dan paket TGS yang masih dalam keadaan terenkripsi. *Client* tidak dapat membuka paket ini karena kunci yang dipakai adalah  $K_{TGS}$ , yang hanya diketahui oleh AS dan TGS. Selanjutnya *client* akan membuat *Authenticator* (Auth atau  $A_C$ ) yang berisi *username*, *IP address client*, dan *time-stamp*. Lalu *client* akan mengirimkan nama *server* yang dituju (S),  $A_C$ , dan mem-forward paket TGS dari AS ke TGS melintasi jaringan. Notasi dari pengiriman tersebut adalah sebagai berikut:

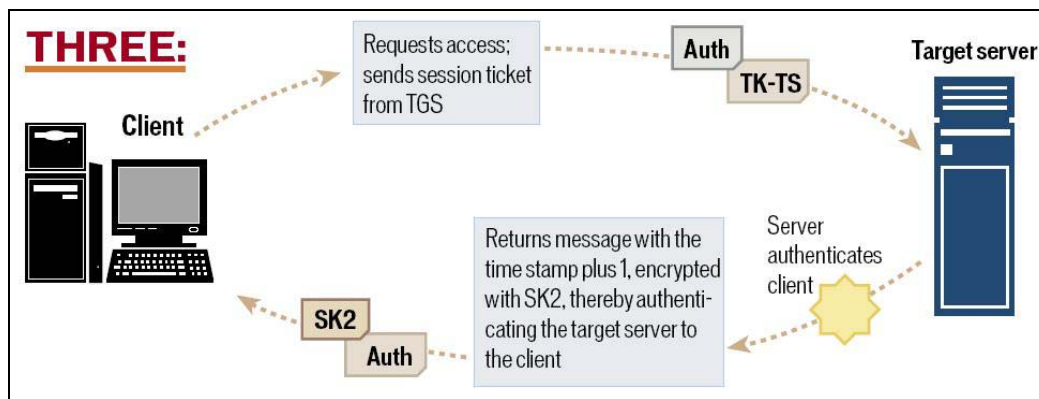
$$S, \{ A_C \}_{K_{C,TGS}}, \{ K_{C,TGS}, T_{C,TGS} \}_{K_{TGS}}$$

Di TGS paket TGS dari AS didekrip dengan  $K_{TGS}$  dan TGS memperoleh  $K_{C,TGS}$  dan  $T_{C,TGS}$ .  $K_{C,TGS}$  digunakan untuk mendekrip  $A_C$ . Jika isi  $A_C$  dan  $T_{C,TGS}$  sesuai, maka TGS akan memberi akses dengan cara membuat *Session Key 2* (SK2 atau  $K_{C,S}$ ) untuk komunikasi antara *client* dan *server* yang dituju (disebut juga *Target Server* atau TS). TGS akan mengeluarkan *ticket* baru

(disebut TK-TS atau  $T_{C,S}$ ).  $T_{C,S}$  dan  $K_{C,S}$  akan dienkripsi dengan kunci privat server ( $K_S$ ) menjadi paket TS dari TGS.  $K_{C,S}$  dan paket TS dienkripsi dengan  $K_{C,TGS}$ , kemudian dikirimkan ke *client* melintasi jaringan. Notasi untuk pengiriman ini dinyatakan sebagai berikut:

$$\{ K_{C,S}, \{ K_{C,S}, T_{C,S} \} K_S \} K_{C,TGS}$$

Warna jingga menunjukkan paket TS dari TGS.



Gambar 2-4. Tahap 3: Client/Server Exchange

Tahap ketiga disebut *Client/Server Exchange*. Pada tahap ini *client* dan *server* yang bersangkutan akan melakukan otentikasi. Otentikasi ini dapat berlangsung searah atau dua arah (*mutual authentication*). Otentikasi searah berarti *client* harus membuktikan ke *server* siapa dirinya, sedangkan pada otentikasi dua arah *server* juga harus membuktikan kepada *client* siapa dirinya. *Client* mendekripsi data yang diterima dengan  $K_{C,TGS}$  dan mendapatkan  $K_{C,S}$  dan paket TS dari TGS. Paket TS ini tidak dapat dibuka oleh *client* karena proses dekripsi dilakukan dengan menggunakan kunci privat  $K_S$  yang hanya diketahui oleh TGS dan TS. Kemudian *client* akan membuat  $A_C$ , dan mengenkripsinya dengan  $K_{C,S}$ . Selanjutnya *client* mengirimkan  $A_C$  tersebut

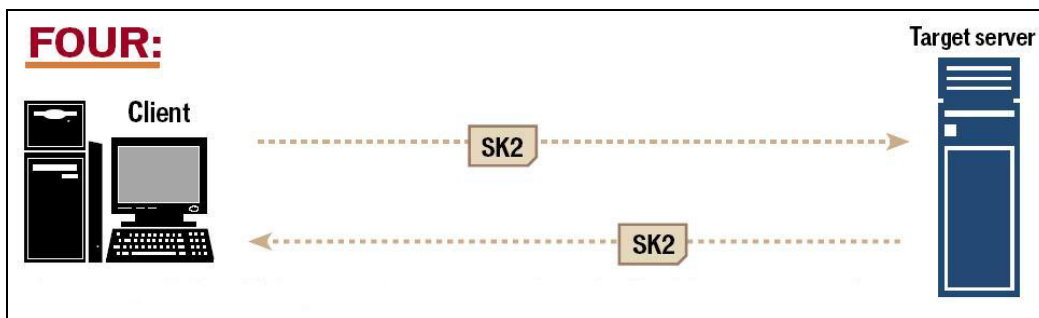
dan mem-*forward* paket TS dari TGS ke *server* yang dituju melintasi jaringan.  
Notasi untuk proses ini dinyatakan sebagai berikut:

$$\{ A_C \}_{K_{C,S}}, \{ K_{C,S}, T_{C,S} \}_{K_S}$$

Sesampainya di TS, *server* akan mendekrip paket TS dari TGS dengan kunci privat yang dimilikinya, dan mendapatkan  $K_{C,S}$  dan  $T_{C,S}$ .  $K_{C,S}$  digunakan untuk mendekrip  $A_C$ . Jika isi  $A_C$  dan  $T_{C,S}$  sesuai, maka TS akan memberi akses kepada *client* untuk mendapatkan *service* darinya. Dengan demikian TS telah diyakinkan bahwa *user* yang meminta *service* padanya adalah *user* yang sah. Jika *mutual authentication* dibutuhkan, maka TS akan mengirimkan data *time-stamp* yang tercantum di  $A_C$  ditambah satu, lalu dienkrpsi dengan *session key*  $K_{C,S}$ . Notasinya adalah sebagai berikut:

$$\{ \text{time-stamp } A_C + 1 \}_{K_{C,S}}$$

Dengan demikian kedua belah pihak diyakinkan akan kebenaran identitas masing-masing.

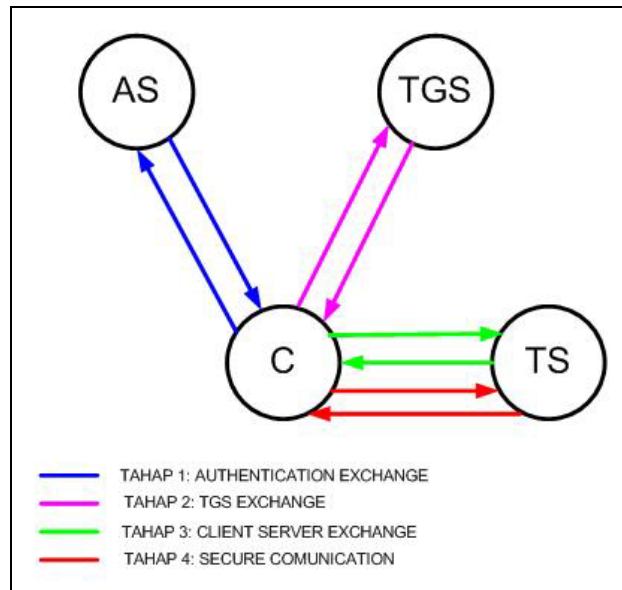


Gambar 2-5. Tahap 4: Secure Communication

Tahap terakhir disebut *Secure Communication*. Baik *client* dan TS telah diyakinkan akan kebenaran identitas masing-masing. Pertukaran data

diantara keduanya dapat dilakukan dengan aman karena *client* dan *server* memiliki kunci privat  $K_{C,S}$  yang hanya diketahui oleh mereka saja.

Gambar 2-6 memperlihatkan jalannya otentikasi yang harus dilalui sebelum *client* mendapat akses ke *server*.



Gambar 2-6. Protokol Kerberos

## 2.6 Keunggulan, Keterbatasan, dan Kelemahan Kerberos

Beberapa makalah telah menunjukkan keterbatasan dan kelemahan dari sistem otentikasi ini. Kerberos bukanlah sebuah sistem yang menjadi solusi untuk semua masalah keamanan jaringan. Berikut ini akan dibahas beberapa keunggulan, keterbatasan, dan kelemahannya.

### 2.6.1 Keunggulan

Keunggulan utama yang dimiliki Kerberos adalah tingkat keamanannya yang tinggi. *Username* dan *password* tidak dikirimkan melintasi jaringan. Hal ini merupakan perbaikan dari sistem konvensional (*password-based*) yang rentan terhadap *eavesdropping attack*.

Selain itu, Kerberos memiliki sifat *transparent*. Artinya, *user* tidak perlu mengetahui tentang tahap-tahap otentikasi yang dilakukan di dalam jaringan. Yang dilakukan *user* hanyalah *login* ke jaringan melalui program inisialisasi kinit, memasukkan *username* dan *password*, dan *user* memperoleh otentikasi ke *server* yang dituju.

### 2.6.2 Keterbatasan

Dalam [6] dikatakan bahwa sistem otentikasi ini didesain untuk menjawab kebutuhan dari Project Athena. Oleh karena itu, asumsi-asumsi yang digunakan disesuaikan dengan keadaan jaringan Project Athena. Lingkungan dalam Project Athena dapat digambarkan sebagai berikut:

- *workstation* anonim dalam jumlah yang besar
- mesin *server* otomatis berukuran besar yang jumlahnya relatif sedikit
- *service-service* yang diperlukan *user* (misalkan penyimpanan *file*) dilakukan di *server*, bukan di *workstation*

Jika Kerberos hendak digunakan menjadi standar, maka asumsi-asumsi yang digunakan harus menjadi lebih umum untuk mengakomodasi lingkungan-lingkungan yang bervariasi. Generalisasi ini akan menambah kemungkinan lubang-lubang keamanan. Jadi, pada lingkungan yang berbeda dari Project Athena Kerberos dapat tidak bekerja seperti yang diharapkan.

### 2.6.3 Kelemahan

Kelemahan di sini diartikan sebagai lubang-lubang keamanan pada protokol ini. Beberapa diantaranya, yaitu:

- *Clock synchronization service* yang aman

Kerberos menggunakan *time-stamp* untuk mengetahui apakah *authenticator* yang dikirimkan masih baru (bukan *replay attack*). Untuk itu dibutuhkan sinkronisasi waktu di seluruh jaringan. Program sinkronisasi waktu yang digunakan umumnya adalah `ntpd`. Namun, seperti halnya *service* lain di jaringan, *service* ini juga memerlukan otentikasi. Jika tidak, maka dapat terjadi lubang keamanan dalam bentuk *replay attack*.

- *Trojan horse attack*

Terjadi jika *workstation* sudah tidak aman lagi. Jika penyerang memodifikasi program dimana *user* memasukkan *username* dan *passwordnya*, maka penyerang dapat memperoleh informasi yang cukup untuk melakukan *impersonation attack* pada sistem ini. Oleh karena itu, Kerberos menuntut jalur yang aman antara *user* dan *workstation*.

- “*Kerberizing*” *application program / client / server*

Program-program aplikasi yang menggunakan otentikasi Kerberos harus diubah *source code*-nya (*di-kerberized*) sehingga dapat berkomunikasi dengan *library-library* Kerberos. Hal ini menjadi masalah sehubungan ukuran dan desain dari program aplikasi, khususnya pada program yang *source code*-nya tidak dipublikasikan. Tidak hanya

program, semua *client / server* dalam jaringan tersebut harus di-*kerberized*. Pilihannya hanya satu: di-*kerberized* atau tidak digunakan sama sekali.

Lubang-lubang keamanan lain yang ditemukan pada Kerberos dan beberapa solusi yang diusulkan untuk mengatasinya dijelaskan secara detail pada referensi [6].

## BAB III

### OTENTIKASI ANTARJARINGAN KERBEROS

#### 3.1 Pendahuluan

Pada jaringan-jaringan yang berbeda, *service-service* yang tersedia tidak seragam. Diinginkan agar *user* pada satu jaringan dapat menggunakan *service* pada jaringan yang lain. Namun, otentikasi tetap menjadi masalah karena baik *user* di jaringan yang satu maupun *server* di jaringan yang lain perlu meyakini bahwa identitas keduanya sah.

Di dalam terminologi Kerberos, hal di atas disebut juga *cross-realm authentication*. *Realm* itu sendiri merupakan istilah untuk jaringan dengan sebuah *server* Kerberos sebagai sistem otentikasinya. Sebuah *realm* memiliki *principal-principal* yang harus dapat dikenali secara unik. Untuk itu, setiap *principal* perlu memiliki nama yang unik.

#### 3.2 Aturan Penamaan Principal

Format nama *principal* pada sebuah realm ditunjukkan seperti di bawah ini:

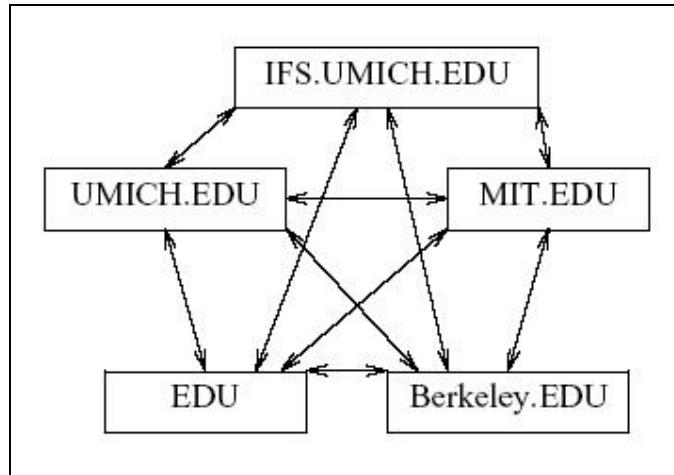
name.instance@realm

**name** menunjukkan nama dari *principal* (*user* atau *server*). **Instance** digunakan jika ada variasi pada nama yang sama. Sedangkan **realm** adalah nama jaringan Kerberos yang bersangkutan.

Sebagai contoh, misalkan terdapat tiga *principal* dalam *realm* bernama ICDESIGN.PAU.ITB, yaitu *user* ivan, *workstation* ic00 dan ic01 yang masing-masing menjadi *server rlogin*. Jika *user* ivan dapat dua buah *login* ke jaringan, yaitu sebagai *root* atau sebagai *admin*, maka *root* dan *admin* ini akan menjadi *instance* dari *user* ivan (yaitu `ivan.root@ICDESIGN.PAU.ITB` atau `ivan.admin@ICDESIGN.PAU.ITB`). Pada *service rlogin*, ic00 dan ic01 akan menjadi *instance* yang membedakan *service rlogin* pada *workstation* yang berbeda (yaitu `rlogin.ic00@ICDESIGN.PAU.ITB` untuk *rlogin* di *workstation* ic00 dan `rlogin.ic01@ICDESIGN.PAU.ITB` untuk *rlogin* di *workstation* ic01).

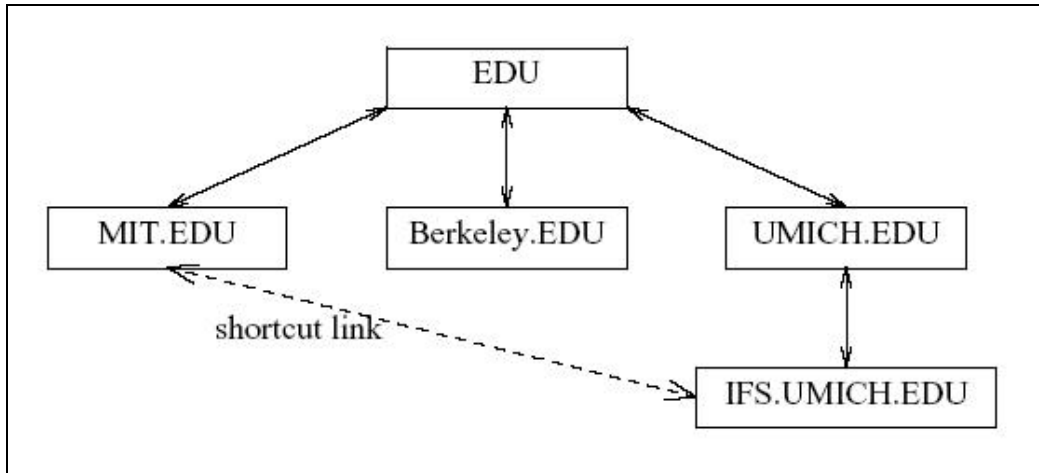
### 3.3 Prinsip Kerja dari *Cross-Realm Authentication*

Seperti halnya pada sistem otentikasi Kerberos dalam satu jaringan, masing-masing *principal* dapat berkomunikasi dengan aman dengan cara men-*share* kunci privat yang sama. Hal yang sama dilakukan juga untuk melakukan otentikasi antar*realm*. Agar sebuah *realm* dapat berkomunikasi dengan *realm* lain dengan aman, keduanya harus men-*share* kunci privat yang sama.



Gambar 3-1. Model Cross-Realm Authentication pada Kerberos V4  
(sumber: [4])

Kerberos Versi 4 telah memiliki kemampuan *cross-realm authentication*. Kedudukan dari *realm-realm* tersebut adalah sejajar (horisontal). Tetapi jika jumlah *realm* yang berkomunikasi bertambah, maka jumlah kunci privat yang harus disimpan untuk komunikasi antar*realm* juga bertambah secara linier (tidak *scalable*). Gambar 3-1 menunjukkan model otentikasi antar*realm* pada Kerberos Versi 4, dimana terdapat 5 *realm* yang akan saling berkomunikasi. Agar *realm* EDU dapat menggunakan *service* yang dimiliki *principal* dari *realm* MIT.EDU, kedua *realm* harus men-*share* kunci privat (ditunjukkan oleh garis dengan tanda panah). Demikian juga untuk *realm* lainnya. Jika terdapat  $n$  *realm*, maka setiap *realm* harus menyimpan  $(n-1)$  kunci privat. Hal ini menjadi masalah, karena jumlah kunci privat yang harus disimpan menjadi tidak *scalable*.



Gambar 3-2. Model Cross-Realm Authentication pada Kerberos V5  
(sumber: [4])

Masalah *scalability* ini mendapat perbaikan pada Kerberos Versi 5. Dengan membagi kedudukan *realm-realm* dalam sebuah hirarki (vertikal), jumlah kunci privat yang harus disimpan oleh setiap *realm* berkurang. Setiap *realm* cukup menyimpan kunci privat dari *realm* anak dan *realm* induknya saja. Dengan demikian, untuk jumlah total  $n$  *realm* yang berkomunikasi, jumlah kunci privat yang harus disimpan tiap *realm* sebanding dengan  $\log(n)$ . Gambar 3-2 adalah contoh dari otentikasi antar*realm* dengan sistem hirarki. *Realm-realm* yang sama kini dibagi dalam hirarki, dimana *realm* EDU mempunyai 3 buah *realm* anak, yaitu MIT.EDU, Berkeley.EDU, dan UMICH.EDU. *Realm* UMICH.EDU cukup menyimpan kunci privat dengan *realm* induk dan *realm* anak saja, yaitu *realm* EDU dan *realm* IFS.UMICH.EDU.

Jika terdapat sejumlah *realm* yang memiliki *traffic* yang padat, maka dapat dibuat *shortcut link* (*sharing* kunci privat) diantara *realm-realm* tersebut. Akibatnya otentikasi antar*realm* tersebut tidak perlu melalui *realm* induknya, tetapi dapat dilakukan langsung (horisontal). Tujuannya adalah untuk

menghindari *bottleneck* di jaringan sehingga tidak terjadi *denial-of-service* (DoS). Pada contoh di atas dapat dimisalkan bahwa *realm* MIT.EDU dan *realm* IFS.UMICH.EDU memiliki *traffic* yang padat, sehingga dibuatlah *shortcut link* berupa *sharing* kunci privat. Dengan demikian otentikasi diantara *realm* MIT.EDU dan *realm* IFS.UMICH.EDU tidak perlu melalui *realm* EDU dan UMICH.EDU yang merupakan *realm* induk.

Dari pembahasan di atas, dapat disimpulkan bahwa struktur jaringan Kerberos dapat dibuat secara vertikal dan horisontal. Untuk mencapai hasil yang optimal perlu dilakukan *trade-off* antara kepadatan *traffic* dan jumlah total kunci privat yang perlu dimiliki setiap *realm* untuk dapat melakukan *cross-realm authentication*.

## BAB IV

### USULAN STRUKTUR JARINGAN KERBEROS DI ITB

#### 4.1 Pendahuluan

Sentralisasi adalah salah satu konsep jaringan yang menggunakan *server* Kerberos sebagai sistem otentikasi. Sebagai contoh, MIT adalah sebuah jaringan besar dengan yang menerapkan Kerberos sebagai sistem otentikasinya. Untuk dapat melayani *authentication request* dari *user* yang sangat banyak, dibuatlah *server-server slave*. Maksudnya *server slave* akan melayani *authentication request* dengan mengacu pada *database* hasil duplikasi dari *database* yang ada pada *server master*. *Database* di *server slave* hanya bersifat *read-only* dan akan di-*update* oleh *server master*. Sementara itu *administration request* (misalnya penambahan atau penghapusan *username*, penggantian *password*, dan data-data administratif lain) hanya dapat diberikan oleh *server master*. Konsep yang sama juga diterapkan di berbagai universitas, seperti University of Michigan dan UCLA Berkeley.

Namun, konsep sentralisasi ini akan menjadi masalah dalam menggunakan Kerberos untuk jaringan komputer yang terdiri atas banyak LAN yang tumbuh secara independen. Salah satu kasusnya adalah jaringan

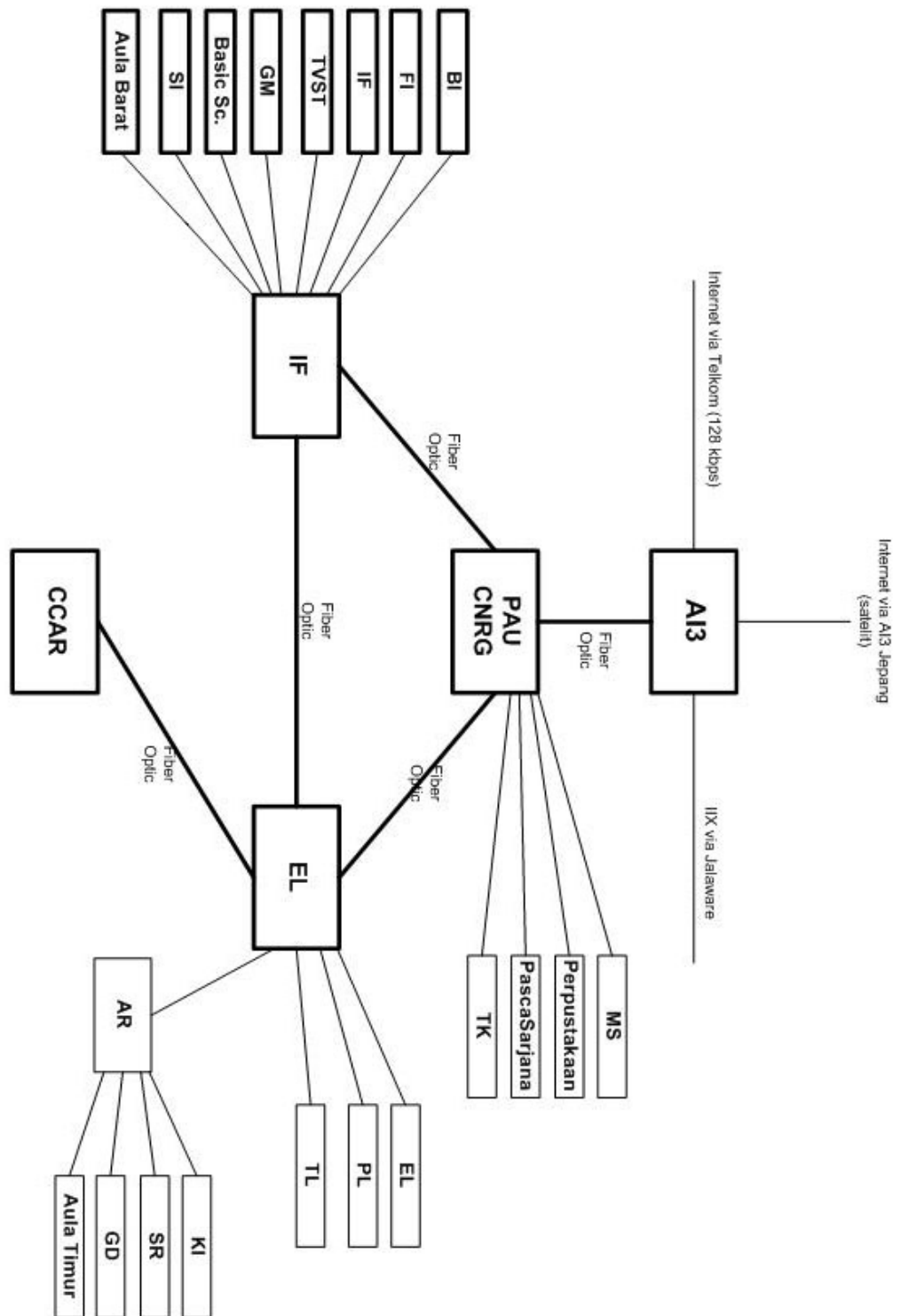
komputer di ITB. Bab ini akan membahas mengenai usulan struktur jaringan Kerberos di ITB.

#### **4.2 Kultur Jaringan Komputer di ITB**

Pada mulanya jaringan komputer ITB terbentuk dari jaringan komputer di departemen-departemen. Jaringan komputer departemen ini tumbuh secara independen satu sama lain, sehingga setiap departemen memiliki regulasi masing-masing. Kemudian AIII ITB masuk sebagai *gateway* internet via satelit yang menghubungkan ITB ke dunia luar. Di samping AIII, ITB juga memiliki koneksi internet via Telkom dan IIX via Jalaware.

Gambar 4-1 menunjukkan peta jaringan komputer ITB secara umum (catatan: ada banyak jaringan komputer departemen yang belum tertera, tetapi peta jaringan ini dirasa cukup untuk memberikan gambaran mengenai struktur jaringan yang akan diusulkan). Koordinasi jaringan komputer ITB dilakukan di empat buah titik, yaitu Gedung PAU (CNRG), Gedung Dept. Teknik Informatika, Gedung Dept. Teknik Elektro, dan Gedung CCAR, dimana instalasi fisiknya menggunakan media *fiber optic*. Secara garis besar jaringan komputer ITB sebelah utara diatur oleh PAU (CNRG), sebelah barat oleh Dept. Teknik Informatika, dan sebelah timur oleh Dept. Teknik Elektro.

Oleh karena kondisi jaringan yang independen, konsep sentralisasi dari Kerberos tidak dapat diterapkan secara langsung seperti halnya Kerberos di tempat lain. Konsep sentralisasi terbentur dengan masalah regulasi jaringan komputer departemen. Untuk itu diperlukan struktur yang membagi jaringan ini menjadi hirarki.



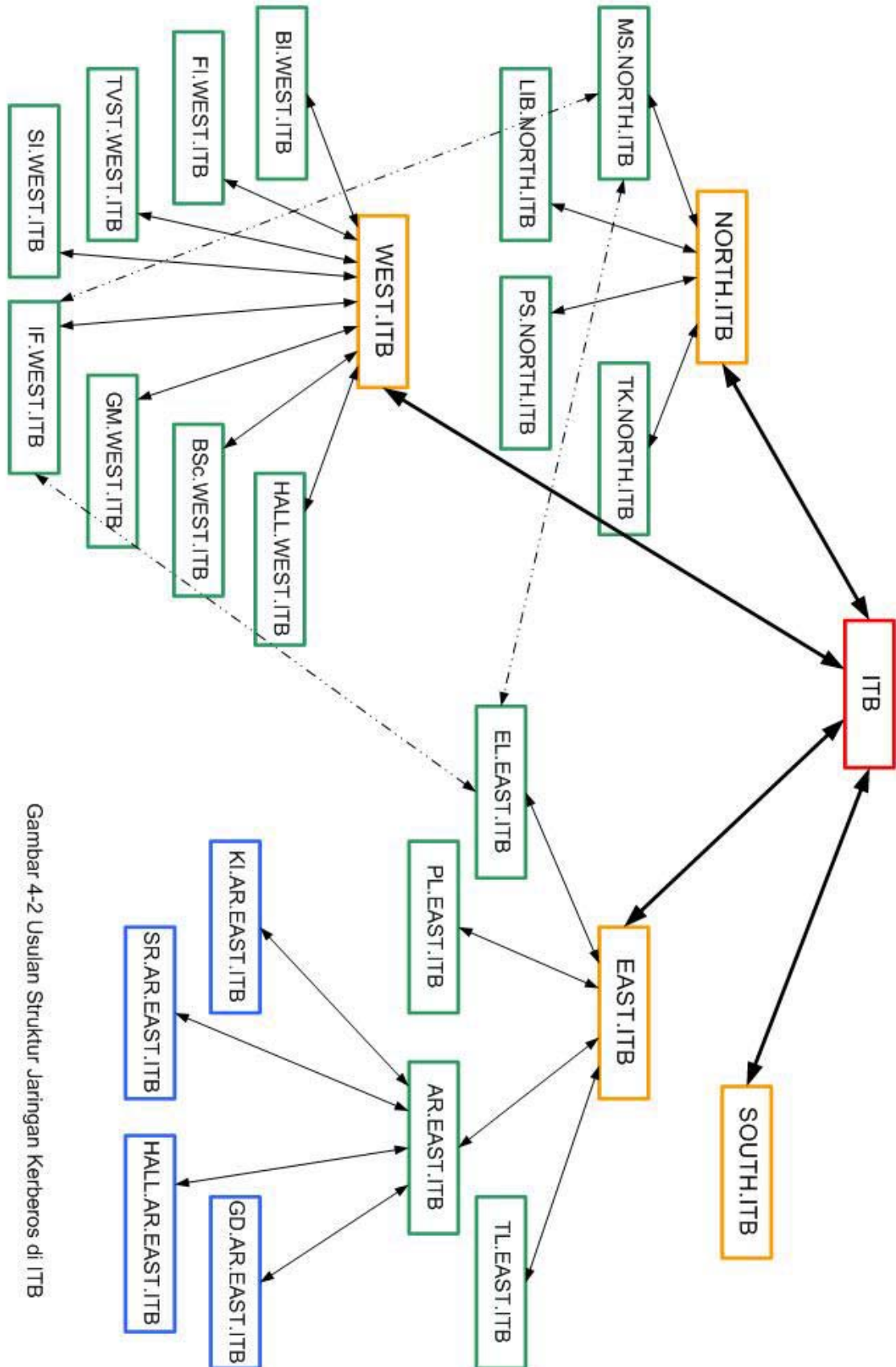
Gambar 4-1 Peta Jaringan Komputer di ITB

### 4.3 Usulan Struktur Jaringan Kerberos di ITB

Agar Kerberos dapat digunakan sebagai sistem otentikasi, jaringan ITB perlu dibagi menjadi *realm-realm* yang terhubung baik secara vertikal maupun secara horisontal. Konsep sentralisasi diterapkan pada jaringan komputer milik departemen. Hal ini dilakukan untuk mengatasi masalah regulasi di tiap departemen. Setiap *realm* departemen akan memiliki *realm* induk. Untuk *realm* departemen – *realm* departemen yang memiliki *traffic* yang padat dapat diberikan *shortcut link*. Gambar 4-2 merupakan usulan struktur jaringan dalam bentuk nama *realm*. Warna yang sama menunjukkan kedudukan *realm* yang sama secara hirarki. Garis putus-putus menunjukkan *shortcut link*.

Misalkan keseluruhan jaringan komputer di ITB menjadi satu *realm* besar dengan nama ITB. Keempat titik koordinasi akan memiliki nama *realm* NORTH.ITB, WEST.ITB, EAST.ITB, dan SOUTH.ITB, berturut-turut untuk PAU (CNRG), Gd. Dept. Teknik Informatika, Gd. Dept. Teknik Elektro, dan CCAR. Setiap jaringan komputer di departemen akan memiliki nama *realm* sesuai departemen, misalkan BI.WEST.ITB untuk Dept. Biologi, MS.NORTH.ITB untuk Dept. Teknik Mesin, TL.EAST.ITB untuk Dept. Teknik Lingkungan, dan sebagainya. *Realm-realm* yang memiliki *traffic* yang padat, misalkan antara *realm* IF.WEST.ITB, *realm* EL.EAST.ITB dan *realm* MS.NORTH.ITB, akan memiliki *shortcut link*. Adapun indikator *traffic* dapat dilihat dari banyaknya *principal* (*user* dan *server*) dalam sebuah *realm*. Dengan demikian otentikasi diantaranya tidak perlu melalui *realm* induk EAST.ITB, *realm* induk ITB, dan *realm* induk NORTH.ITB. Dengan kata lain, ketiga *realm* tersebut berkomunikasi secara horisontal satu dengan yang lain.

Optimasi untuk usulan struktur jaringan ini dapat dilakukan dengan:



Gambar 4-2 Usulan Struktur Jaringan Kerberos di ITB

- menggabungkan beberapa jaringan berukuran kecil ke dalam sebuah *server* Kerberos (misalkan satu *server* Kerberos membawahi 100-200 *principal*).
- memberikan *shortcut link* pada jaringan dengan *traffic* yang padat.

Oleh karena keterbatasan pengetahuan penulis mengenai keadaan sebenarnya dari jaringan komputer di ITB, optimasi tidak dapat dilakukan

#### 4.4 Kendala yang akan Dihadapi

Dapat dikatakan bahwa usulan struktur jaringan di atas hanya membagi jaringan ITB ke dalam hirarki dan memberikan *shortcut link* pada *realm-realm* yang memiliki tingkat *traffic* yang tinggi. Tetapi ada beberapa kendala yang muncul, yaitu:

- Jumlah *server* Kerberos terlalu banyak

Untuk departemen yang memiliki jaringan yang kecil, penggunaan *server* Kerberos tidak efektif. Akan lebih baik jika beberapa jaringan departemen kecil digabung dalam satu *server* Kerberos. Hal ini akan mempermudah pengamanan fisik dari *server* Kerberos (jumlah *server* Kerberos akan lebih sedikit sehingga pengamanan lebih mudah). Namun, tidak semua departemen akan mengizinkan, karena tiap departemen memiliki otoritas dan regulasi sendiri-sendiri atas jaringannya.

- Instalasi program tambahan di setiap *workstation*

Instalasi program Kerberos akan menyita banyak waktu, dikarenakan jumlah *workstation* yang sangat banyak dan tersebar. Tetapi dengan koordinasi yang baik hal ini dapat dilakukan.

- *compatibility* dari protokol Kerberos dengan *operating system* yang ada Jaringan ITB merupakan *heteroculture network*. *Workstation-workstation* yang ada mempunyai *operating system* (OS) yang bervariasi, misalkan UNIX, Linux, Window 98, Window 2000, Window XP, FreeBSD, dsb. Oleh karena itu, harus dicek apakah Kerberos untuk OS tertentu *compatible* dengan Kerberos untuk OS lain.
- *kerberizing* seluruh *service* yang digunakan  
Seperti yang telah disebutkan di Bab 2, *kerberizing* menjadi syarat agar sistem otentikasi ini dapat bekerja dengan baik. Jika ada *service* jaringan yang belum di-*kerberized*, maka perlu usaha tambahan untuk memodifikasi *source codenya*.

Untuk memperoleh sistem otentikasi yang handal, Kerberos juga dapat diintegrasikan dengan sistem otentikasi lain yang telah digunakan sebelumnya. Adapun sistem otentikasi yang kini digunakan di jaringan ITB adalah LDAP. Keuntungan dan kerugian integrasi Kerberos dan LDAP dapat tidak dibahas di makalah ini.

## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan

Dari hasil studi literatur dan wawancara dengan seseorang yang mengerti secara langsung jaringan di ITB, dapat ditarik beberapa kesimpulan sebagai berikut:

- Sistem Otentikasi Kerberos merupakan salah satu solusi untuk mengatasi serangan-serangan keamanan di jaringan yang menjadi kelemahan sistem otentikasi konvensional (*password based*).
- Kerberos melakukan otentikasi dengan men-*share* kunci privat antara *client / server*, dimana kunci privat tersebut dikeluarkan oleh pihak ketiga yang dipercayai bersama.
- Agar Kerberos dapat diterapkan di jaringan komputer ITB, struktur jaringan perlu diubah untuk memenuhi konsep sentralisasi dari Kerberos. Perubahan ini berupa pembagian jaringan komputer ke dalam *realm-realm*.
- Untuk optimasi dapat dilakukan *merger* beberapa jaringan ke dalam sebuah *server* Kerberos, dan pemberian *shortcut link* untuk *realm* dengan *traffic* tinggi.

- Perlu usaha ekstra untuk mengubah sistem otentikasi sebelumnya dengan Kerberos. Usaha ini, antara lain, berupa instalasi Kerberos di *workstation*, *kerberizing service-service* jaringan, dan mengecek *compatibility*-nya dengan OS yang ada.

## REFERENSI

- [1] B. C. Neuman, T. Ts'o. *Kerberos: An Authentication Service for Computer Networks*. IEEE Communications, 32(9): 33-38, September 1994.
- [2] B. Tung. *The Moron's Guide To Kerberos, Version 1.2.2*. <http://www.isi.edu/gost/brian/security/kerberos.html>, December 19 1996.
- [3] J. G. Steiner, C. Neuman, and J. I. Schiller. *Kerberos: An Authentication Service for Open Network Systems*. Massachusetts Insitute of Technology, January 12 1988.
- [4] J. T. Kohl, B. C. Neuman, T. Ts'o. *The Evolution of The Kerberos Authentication Service*. EurOpen Conference (revision), Spring 1991.
- [5] M. A. Sirbu, J. C. Chuang. *Distributed Authentication in Kerberos Using Public Key Cryptography*. Carnegie Mellon University.
- [6] S. M. Bellovin, M. Merrit. *Limitation of The Kerberos Authentication System*. AT&T Bell Laboratories. Computer Communication Review, 20(5):119-132, October 1990.
- [7] S. P. Miller, B. C. Neuman, J. I. Schiller, J. H. Saltzer. *Kerberos Authentication and Authorization System*. Project Athena Technical Plan Section E.2.1, Massachusetts Insitute of Technology, October 27 1988.
- [8] <http://www.computerworld.com/computerworld/records/images/pdf/>, *Sharing a Secret: How Kerberos Works*.
- [9] <http://www.stanford.edu/services/kerberos/>, *Kerberos at Stanford*

- [10] <http://web.mit.edu/kerberos/www/dialogue.html>, *Designing an Authentication System: A Dialogue in Four Scenes*, February 1997.
- [11] Red Hat Enterprise Linux 3: Reference Guide, *Chapter 18. Kerberos*
- [12] <http://nic.itb.ac.id>