Pemanfaatan Graf pada Manajemen Kunci Kriptografi untuk Mengamankan Komunikasi Grup

Annisa Dini Handayani

Sekolah Tinggi Sandi Negara
annisa.dini@stsn-nci.ac.id

Abstrak

Komunikasi grup merupakan salah satu bentuk komunikasi yang mengijinkan *user* untuk mentransmisikan data kepada *n user* lain dalam satu grup komunikasi. Permasalahan utama dari komunikasi grup ini adalah keamanan dari data yang ditransmisikan. Meskipun enkripsi dapat digunakan untuk mengamankan data tersebut, tetapi manajemen kunci kriptografi yang digunakan untuk proses enkripsi masih menjadi permasalahan yang perlu diperhatikan. Pada makalah ini akan dibahas bagaimana graf pohon dapat digunakan untuk memodelkan bentuk komunikasi grup sehingga akan memudahkan manajemen kunci enkripsi yang digunakan pada komunikasi grup tersebut. Dengan memanfaatkan graf pohon, pembaruan kunci ketika terjadi penambahan dan pengurangan user akan menjadi lebih mudah.

Kata Kunci – komunikasi grup, manajemen kunci, kriptografi, graf pohon.

PENDAHULUAN

Komunikasi grup merupakan sistem komunikasi yang menyediakan transmisi data dari sejumlah titik ke sejumlah titik lainnya menggunakan suatu proses dalam suatu grup (Chockler *et al*., 2001). Dalam suatu jaringan komputer terdistribusi, komunikasi grup dapat bersifat *multicast* (*one-to-many communication*), yaitu mekanisme transmisi data dari satu titik sumber (*source*) ke sejumlah titik tujuan (*destinations*). Bentuk khusus dari *multicast* adalah *unicast* (*one-to-one communication*) dan *broadcast* (*one-to-all communication*). *Unicast* merupakan mekanisme transmisi data dari satu titik sumber ke satu titik tujuan, sedangkan *broadcast* merupakan mekanisme transmisi data sari satu titik sumber ke semua titik tujuan dalam suatu grup (Liang *et al*., 1990).

Hal yang menjadi permasalahan utama dalam komunikasi *multicast* adalah otentikasi dan kendali akses *user* serta keamanan data yang ditransmisikan. Salah satu metode yang dapat digunakan untuk membatasi akses terhadap informasi/data yang ditransmisikan adalah enkripsi. Proses enkripsi memerlukan suatu algoritma enkripsi, yaitu algoritma yang memerlukan *input* berupa data/informasi yang selanjutnya akan ditransformasikan menjadi suatu *ciphertext* menggunakan kunci kriptografi. *User* yang tidak mengetahui kunci kriptografi ini tidak bisa membaca *ciphertext* yang ditransformasikan, sehingga metode enkripsi dapat memberikan keamanan data pada komunikasi *multicast*. Dalam konteks komunikasi grup, kunci kriptografi ini disebut sebagai kunci grup (*group key*).

Terdapat permasalahan yang timbul saat mengimplementasikan metode enkripsi, yaitu manajemen kunci grup yang digunakan pada komunikasi *multicast*. Manajemen kunci grup memiliki peranan yang sangat penting dalam pengaturan kunci grup, mulai dari proses pembangkitan, distribusi, sampai penghancuran kunci grup. Manajemen kunci grup juga membahas teknik dan prosedur identifikasi dan otentikasi serta kendali akses anggota grup. Masalah utama manajemen kunci pada komunikasi grup adalah ketika terjadi penambahan dan pengurangan *user* dalam grup tersebut. Manajemen kunci harus menyediakan mekanisme pembaruan kunci sehingga terjamin *backward secrecy* dan *forward secrecy* ketika terjadi penambahan dan pengurangan *user*.

Pada makalah ini, akan dibahas manajemen kunci yang dapat digunakan untuk komunikasi grup. Manajemen kunci grup ini akan memanfaatkan graf pohon (*tree*) untuk

memodelkan bentuk komunikasi grup. Dengan pemanfaatan graf pohon, manajemen kunci, khususnya proses distribusi dan pembaruan kunci akan menjadi lebih mudah..

PEMBAHASAN

Pembahasan dalam makalah ini terbagi menjadi 5 (lima) bagian. Bagian 1 (satu) dibahas konsep dasar dari graf dan graf pohon. Konsep dasar manajemen kunci dibahas pada bagian 2 (dua), sedangkan hasil pengkajian dijelaskan pada bagian 3 (tiga).

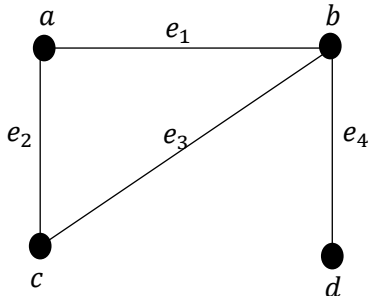
1. Graf dan Pohon (*Tree*)

Pada bagian ini akan dibahas konsep dasar dari graf dan graf pohon

Definisi 1

Suatu graf $G = (V, E)$ terdiri dari V, yang merupakan himpunan tidak kosong dari simpul (*vertices* atau *nodes*), dan E, yang merupakan himpunan dari sisi (*edges*). (Rosen, 2012)

Contoh:



Gambar 1. Graf G

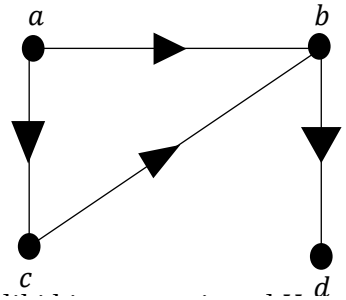
Graf G pada gambar 1 memiliki himpunan simpul $V = \{a, b, c, d\}$ dan himpunan sisi $E = \{e_1, e_2, e_3, e_4\}$.

Sisi pada suatu graf dapat memiliki arah dan tidak memiliki arah. Bila semua sisi pada suatu graf memiliki arah, maka graf tersebut merupakan graf berarah (*directed graph*). Sebaliknya, jika semua sisi pada suatu graf tidak memiliki arah, maka disebut graf tidak berarah (*undirected graph*).

Definisi 2

Suatu graf berarah (*directed graph* atau *digraph*) $G = (V, E)$ terdiri dari himpunan tidak kosong dari simpul, V, dan himpunan sisi berarah/ busur (*arcs*), E. Setiap busur berasosiasi dengan pasangan terurut dari simpul. Suatu busur yang berasosiasi dengan pasangan terurut (u, v) berarti busur tersebut memiliki titik awal simpul u dan titik akhirnya adalah simpul v. (Rosen, 2012)

Contoh:



Gambar 2. Graf Berarah G

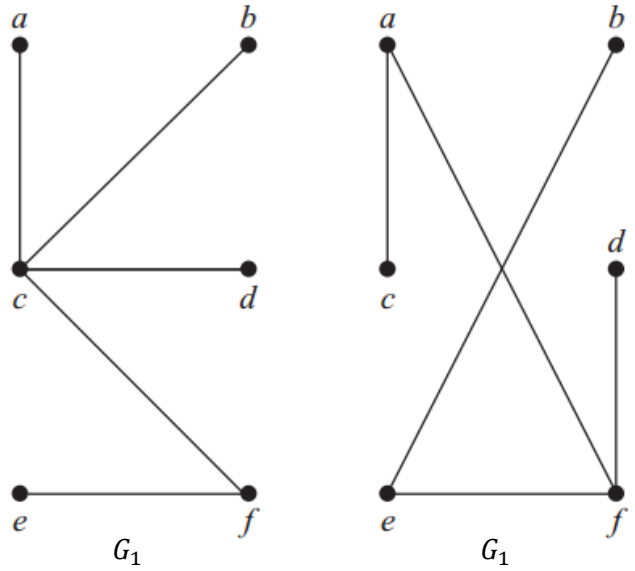
Graf G pada gambar 2 mer… $\{a, b, c, d\}$ dan himpunan sisi $E = \{(a, b), (a, c), (b, d), (c, b$…

Tidak semua pasangan simpul pada suatu graf memiliki sisi. Jika setiap pasangan simpul berbeda pada suatu graf dihubungkan oleh suatu sisi, maka graf tersebut merupakan graf terhubung (*connected graph*).

Definisi 3
Suatu graf tidak berarah disebut terhubung (*connected*) jika terdapat sisi untuk setiap pasangan simpul yang berbeda pada graf tersebut. (Rosen, 2012)

Salah satu tipe khusus dari graf adalah pohon (*tree*). Pohon banyak digunakan dalam ilmu komputer, misalnya dalam penentuan pengkodean Huffman (*Huffman Coding*)

Definisi 4
Pohon merupakan graf tidak berarah terhubung (*connected undirected graph*) yang tidak memiliki sirkuit. (Rosen, 2012)

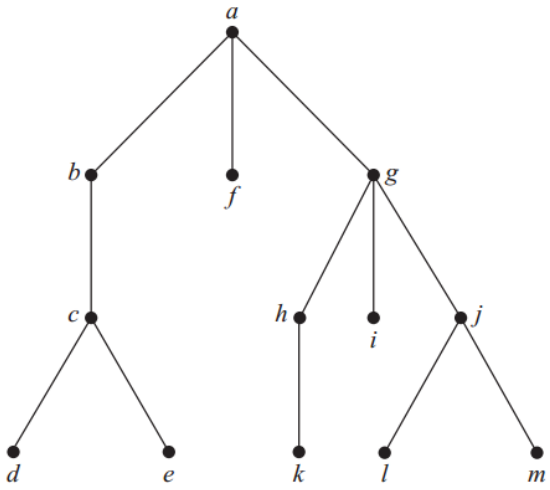


Gambar 3. Contoh Pohon

Dalam aplikasinya, sebuah simpul dari pohon ditetapkan menjadi akar (*root*). Sebuah pohon yang telah memiliki akar disebut *rooted tree*.

Definisi 5
Rooted tree T merupakan suatu pohon yang salah satu simpulnya telah ditetatapkan menjadi akar dan setiap sisi bermula dari akar tersebut. (Rosen, 2012)

Jika v merupakan simpul dari T selain dari akar, maka orang tua (*parent*) dari v adalah simpul unik u sedemikian sehingga terdapat sisi dari u ke v. Jika u adalah orang tua dari v, maka v disebut anak (*child*) dari u. Suatu simpul dari *rooted tree* disebut daun (*leaf*) jika simpul tersebut tidak memiliki anak.



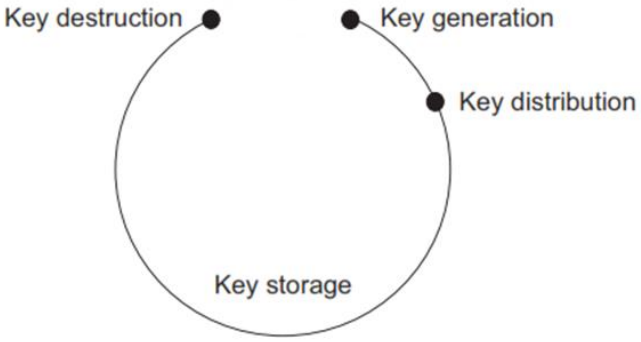
Sumber: Rosen, 2012.

Gambar 4. *Rooted Tree T* dengan akar *a*

2. Manajemen Kunci Kriptografi

Pada bagian ini dibahas konsep dasar dari manajemen kunci kriptografi beserta tahapannya. Menurut NIST *Special Publication* 800-57, manajemen kunci diartikan sebagai aktifitas yang menangani kunci kriptografi dan parameter keamanan lainnya selama masa siklus hidupnya, yang meliputi pembangkitan, penyimpanan, penetapan, *entry* dan *output*, penggunaannya hingga pemusnahannya. Dari pengertian ini, maka manajemen kunci memiliki beberapa tahapan dalam pengelolaan kunci, mulai dari pembangkitan sampai pemusnahan kunci.

Menurut Oppliger, siklus hidup kunci kriptografi yang paling sederhana terdiri dari pembangkitan kunci (*key generation*), distribusi kunci (*key distribution*), penyimpanan kunci (*key storage*) dan pemusnahan kunci (*key destruction*). Pembangkitan kunci merupakan proses untuk menghasilkan kunci menggunakan teknik-teknik tertentu. Pendistribusian kunci merupakan proses distribusi kunci dari satu entitas kepada entitas lain yang memiliki kewenangan. Penyimpanan kunci merupakan proses pengamanan kunci sehingga kunci tersebut terjamin keamanannya selama masa penyimpanan sedangkan pemusnahan kunci merupakan proses penghancuran kunci sehingga kunci tersebut tidak bisa didapatkan kembali dengan cara apapun. Siklus hidup kunci sederhana ini diilustrasikan pada Gambar 5.



Sumber: Oppliger, 2005.

Gambar 5 Siklus Hidup Kunci Kriptografi Sederhana

Pada implementasinya, terdapat kondisi yang membutuhkan proses tambahan selain keempat tahapan di atas. Misalkan jika kunci yang telah didistribusikan hilang, maka diperlukan mekanisme pembaruan kunci baru (*re-keying*). Pembaruan kunci ini dapat dilakukan pada saat kunci lama sudah mendekati akhir dari masa berlakunya, atau kunci yang digunakan saat ini sudah diketahui pihak lain yang tidak berhak. Pada komunikasi grup, pembaruan kunci dapat dilakukan ketika terjadi penambahan atau pengurangan *user* pada grup tersebut.

Manajemen kunci pada komunikasi grup dapat diklasifikasikan menjadi tiga, yaitu:

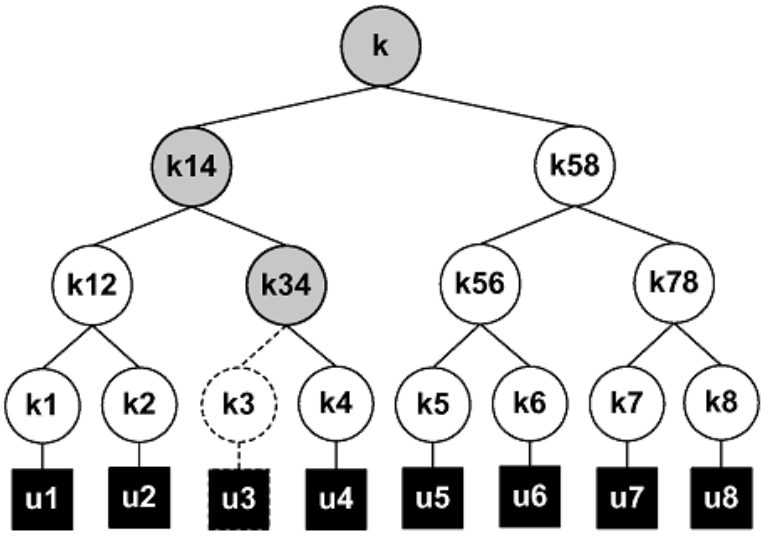
a. Protokol manajemen kunci grup terpusat (*Centralized group key management protocols*);
   Pada tipe ini, terdapat satu entitas yang menjadi pusat kendali dari manajemen kunci kriptografi. Semua fungsi manajemen kunci dilakukan oleh entitas tersebut.
b. *Decentralized architectures*;
   Pada tipe ini, fungsi manajemen kunci dibagi menjadi beberapa subgrup, sehingga meminimalisir kesalahan jika dilakukan secara terpusat.
c. Protokol manajemen kunci terdistribusi (*Distributed key management protocols*).
   Pada tipe ini, *user* memiliki kewenangan untuk melakukan fungsi manajemen kunci, seperti pembangkitan kunci dan distribusi kunci.

3. Hasil Pengkajian

Pada bagian ini dibahas hasil pengkajian berupa pemanfaatan pohon pada manajemen kunci grup terpusat, dengan model *Logical Key Hierarchy*.

Manajemen kunci grup terpusat dengan pendekatan *Logical Key Hierarchy* memanfaatkan entitas pusat yaitu *Key Distribution Centre* (KDC) sebagai pusat kendali pohon kunci. Setiap simpul dari pohon menandakan kunci enkripsi, sedangkan daun dari pohon menandakan anggota grup dan setiap anggota grup tersebut memiliki *Key Encrypting Key* (KEK) yang berkorespondensi dengan dirinya sendiri. Kunci yang dimiliki oleh akar merupakan kunci grup. Untuk pohon yang seimbang, setiap anggota grup memiliki paling banyak $(\log_2 n) + 1$ kunci, dengan $(\log_2 n)$ merupakan tinggi pohon dan n adalah banyaknya *user*.

Pada Gambar 6, notasi $u_1, u_2, \dots, u_8$ menandakan *user1*, *user2*, ..., *user8*. Karena $n = 8$, maka setiap *user* akan memiliki $(\log_2 8) + 1 = 4$ kunci. Misalkan, *user1* akan memiliki 4 kunci, yaitu k_1, k_{12}, k_{14}, dan k; *user5* juga memiliki 4 kunci, yaitu k_5, k_{56}, k_{58}, dan k. Kunci grup adalah k, yang merupakan kunci yang dimiliki oleh akar. Kunci k ini dapat digunakan untuk komunikasi grup yang melibatkan semua *user*, sedangkan kunci k_{ij} merupakan kunci grup untuk komunikasi yang melibatkan *useri*, *user*$(i + 1)$, ..., *userj*. Misalkan kunci k_{14} dapat digunakan untuk komunikasi grup antara *user1*, *user2*, *user3*, dan *user4*.



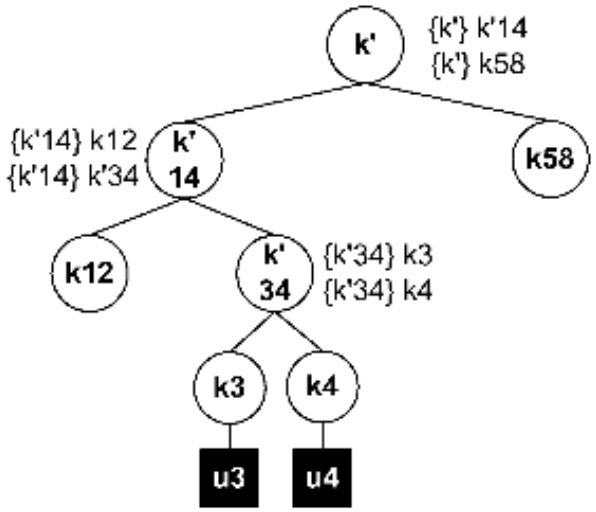
Sumber: Rafaeli dan Hutchison, 2003.

Gambar 6. Contoh Pohon Representasi dari Kunci Grup.

Permasalahan yang biasa terjadi pada komunikasi grup adalah penambahan *user* ataupun pengurangan *user* (*user* meninggalkan grup). Kondisi tersebut pasti akan mempengaruhi manajemen kunci grup yang sudah ada. Baik penambahan *user* ataupun pengurangan *user* akan membutuhkan kunci grup baru yang digunakan untuk komunikasi grup selanjutnya.

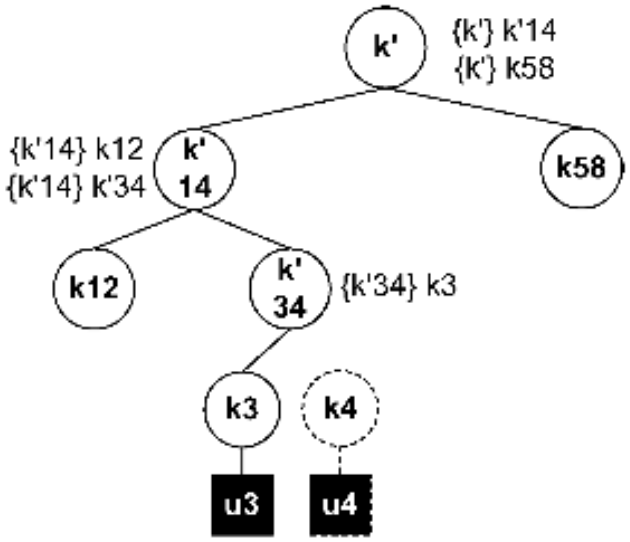
Penambahan *user* akan membutuhkan pembangkitan kunci grup baru. Hal ini dikarenakan jika tidak ada pembangkitan kunci baru, maka *user* yang baru dapat mengetahui informasi yang dikomunikasikan sebelumnya (*backward secrecy*). Misalkan pada Gambar 6, *user3* masuk menjadi *user* baru. Kunci yang harus diperbarui adalah k, k_{14}, k_{34} dan juga harus dibangkitkan kunci baru yaitu kunci k_3. Mekanisme pembaruan kunci ini dilakukan oleh KDC. KDC akan membangkitkan kunci $k', k'_{14}, k'_{34}, k_3$. Proses pengamanannya juga akan berbeda untuk masing-masing kunci baru tersebut. Pada saat masuk menjadi *user* baru, *user3* akan menerima secara langsung kunci k_3 dari KDC. Selanjutnya, KDC akan mengenkripsi kunci k' menggunakan kunci k'_{14} dan kunci k_{58} yang dinotasikan dengan $(k')_{k'_{14}}, (k')_{k_{58}}$. Untuk kunci k'_{14} akan dienkripsi menggunakan kunci k_{12} dan k'_{34}, yaitu $(k'_{14})_{k_{12}}, (k'_{14})_{k'_{34}}$, sedangkan kunci k'_{34} akan dienkripsi menggunakan k_3 dan k_4 yang dinotasikan dengan $(k'_{34})_{k_3}, (k'_{34})_{k_4}$. Proses Pembaruan dari kunci-kunci ini dapat dilihat pada Gambar 7.

Pengurangan *user* dari pohon kunci juga membutuhkan pembaruan kunci. Hal ini dikarenakan jika tidak dilakukan pembaruan kunci, maka *user* yang keluar dari grup dapat mengetahui informasi yang dikomunikasikan setelah keluar dari grup tersebut (*forward secrecy*). Misalkan pada Gambar 6, *user4* akan meninggalkan grup. Karena *user4* berada pada grup yang sama dengan *user3*, maka kunci yang harus diperbarui adalah k, k_{14}, k_{34}. Sama seperti sebelumnya, maka KDC harus membangkitkan kunci baru, yaitu k', k'_{14}, k'_{34}. Enkripsi pada kunci baru tersebut sama seperti penambahan *user3* yang telah dijelaskan sebelumnya. Hal yang berbeda adalah kunci k'_{34} hanya dienkripsi menggunakan k_3. Proses pembaruan kunci ketika terdapat *user* yang meninggalkan grup ini dapat dilihat pada Gambar 8.



Sumber: Rafaeli dan Hutchison, 2003.

Gambar 7. Enkripsi Kunci ketika terdapat Anggota Baru pada Pohon Kunci



Sumber: Rafaeli dan Hutchison, 2003.

Gambar 8. Enkripsi Kunci ketika terdapat *User* yang Meninggalkan Grup

SIMPULAN

Graf pohon dapat digunakan untuk memodelkan bentuk komunikasi grup. Dengan menerjemahkan bentuk komunikasi grup pada graf pohon akan mempermudah proses manajemen kunci yang digunakan pada komunikasi grup tersebut. Misalnya, dengan melihat graf pohon, maka akan lebih mudah ditentukan banyaknya kunci enkripsi yang harus dibangkitkan dan didistribusikan untuk setiap *user*. Selain itu, proses pembaruan kunci juga akan lebih mudah dilakukan dengan melihat struktur dari graf pohon kunci pada komunikasi grup.

DAFTAR PUSTAKA

Chockler, Gregory V., Keidar, Idit., & Vitenberg, Roman. Group Communication Specifications: A Comprehensive Study. *ACM Computing Surveys* (pp. 1-43).

Elaine, Barker. 2016. *Recommendation for Key Management, Part 1: General*. National Institute of Standards and Technology, Special Publication 800-57 Part 1, Revision 4.

Liang, Luping., Chanson, Samuel T., & Neufeld, Gerald W. 1990. *Institute of Electrical and Electronics Engineers (IEEE)*.

Oppliger, Rolf. 2005. *Contemporary Cryptography*. London: Artech House, Inc.

Rafaeli, Sandro., & Hutchison, David. 2003. A Survey of Key Management for Secure Group Communication. *ACM Computing Surveys*, Vol. 35, No. 3, September 2003 (pp. 309-329).

Rosen, Kenneth. 2012. *Discrete Mathematics and Its Application, Seventh Edition*. New York: McGraw-Hill Companies, Inc.