Information Flow Control in Group Communication *

7K−1

Hiroya Mita and Makoto Takizawa †
Tokyo Denki University ‡
e-mail{mita,taki}@takilab.k.dendai.ac.jp

1 Introduction

In the group communication [1], multiple entities send data units to the destination entities and can receive the data units sent in the group. Many papers [1, 4, 5] have discussed how to provide application entities with the atomic and ordered delivery of the data units to the destinations in the group. In addition to the atomic and ordered delivery, it is important to provide secure group communication, i.e. only and all the proper entities can communicate in the presence of malicious entities. [6] discusses how only and all the proper entities in the group obtain a common secret key by using the public key system in order to provide the *secrecy* and *authenticity* of the group communication.

One entity would not like another entity to receive the data units sent by it in the group. An entity E_i which receives the data unit from E_j may forward the data unit to another entity E_k. Even though E_j would not like to send the data unit to E_k, E_k can obtain it through E_i. It is irregal information flow. In addition to protecting the group from being attacked by the outside, we have to control the flow of information among the entities in the group. [7] discusses the lattice-based information flow model which is a generalization of the multi-level security model.

In this paper, we discuss the data transmission procedure for the group of multiple entities which provides *regal* information flow. Each entity has a security class and some precedence relation is defined over the security classes. Each entity can send data units to only entities which have some security classes preceding the security class of the entity. By using the security class, we would like to present the data transmission procedure for the cluster.

In section 2, we present a model of the communication system. In section 3, we discuss a lattice of security classes In section 4, we present the data transmission procedure on the basis of the security classes.

2 System Model

The communication system is composed of *application*, *system*, and *network* layers [Figure 1]. The network layer provides the system entities with the high-speed communication [3]. The entities at the system layer can communicate with each other by using the network layer to provide the application entities with group communication. While the high-speed network provides high-reliable communication, the system entities may fail to receive data units because the transmission speed of the network is faster than the processing speed of the entities. Each application entity A_i takes the service through the system service access point (SAP) S_i supported by a system entity E_i. A *cluster* C is a set of the system SAPs S_1, ..., S_n, which is an extension of the conventional one-to-one connection among two SAPs. C is referred to as *supported* by E_1, ..., E_n, written as $\langle E_1, ..., E_n \rangle$. E_i is referred to as *support* C.

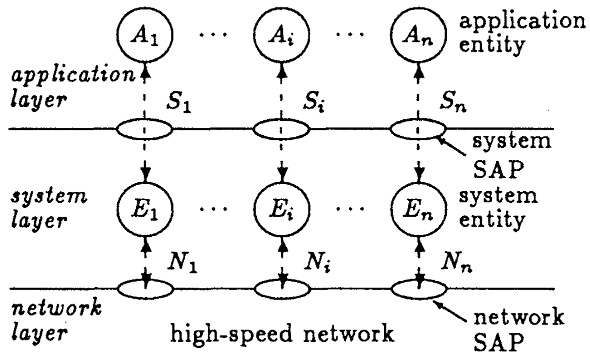

Figure 1: System model

A protocol data unit (PDU) is a unit of communication among peer entities. Each data unit exchanged among system entities consists of the following fields (j = 1,...,n).

- $p.CID$ = cluster identifier.
- $p.SRC$ = entity E_i which transmits p.
- $p.DST$ = set of destination entities of p.
- $p.TSEQ$ = total sequence number of p.
- $p.PSEQ_j$ = partial sequence number for E_j.
- $p.ACK_j$ = total sequence number of a PDU which E_i expects to receive next from E_j.
- $p.BUF$ = number of buffers available in E_i.
- $p.DATA$ = data to be broadcast.
- $p.CLASS$ = security class.

3 Muilti-Level Security

Suppose that cluster entities E_1, ..., E_n. Each entity E_i has one security class $class(E_i)$. Let S be a set of security classes. $\rightarrow \subseteq S^2$ is a partially ordered relation on S. For every pair of security classes s_1 and s_2 in S, the information of the security class s_1 can be flown into the entities of s_2 iff $s_1 \rightarrow s_2$. Here, s_2 is referred to as *dominate* s_1. The partially ordered set $\langle S, \rightarrow \rangle$ is a lattice [2] $\langle S, \rightarrow, \cup, \cap \rangle$, where $\cup$ is a greatest lower bound (glb) and $\cap$ is a lowest upper bound (lub).

[Example] Suppose that there are three entities E_1, E_2, and E_3 in a cluster C whose security classes are s_1, s_2, and s_1, respectively. Suppose that there is a precedence relation $s_1 \rightarrow s_2$. E_1 and E_3 can send the

---

*グループ通信における情報流制御
†三田 浩也, 滝沢 誠
‡東京電機大学

data units to E_2 because $s_1 \rightarrow s_2$, but E_2 can send the data units to neither E_1 nor E_3. Suppose that E_1 sends a data unit p_1 to E_3 and E_2 sends p_2 to E_3 in C. Let s_1, s_2, and s_3 be security classes of E_1, E_2, and E_3, respectively. E_3 can receive data units from both E_1 and E_2 if $s_1 \cup s_2 \rightarrow s_3$. E_3 can send data units to both E_1 and E_2 if $s_3 \rightarrow s_1 \cap s_2$. □

[Definition] The information flow in the cluster C is *regal* if for every data unit p sent to E_j in C, class(S_i) $\rightarrow$ class(S_j). □

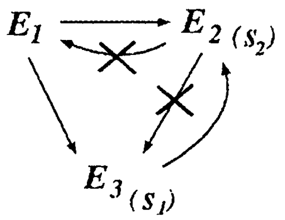

Figure 2: Information flow

4 Roled Cluster

4.1 Roles

We would like to redefine the cluster $C = \langle E_1, ..., E_n \rangle$ to be a tuple of roles $\langle R_1, ..., R_n \rangle$ to take into account the security. Each application entity A_i is *bound* to C with R_i when C is established by the cooperation of $E_1, ..., E_n$. This means that E_i plays a role R_i in C ($i = 1, ..., n$). Each role R_i is defined to be a collection of a security class s_i and a collection O_i of operations for C, i.e. $R_i = \langle S_i, O_i \rangle$. There are the following operations for C, i.e. *send*, *receive*, *abort*, *open*, and *reset* operations. E_i can apply an operation op to C if $op \in O_i$. Suppose that there are three entities A_1, A_2, and A_3. A_1 is bound with a role R_1 which has $O_1 = \{receive\}$. A_1 can only receive data units sent in C while A_i cannot send the data units.

Suppose that each A_i is bound with the role R_i to C. A_i is assigned a security class s_i. Here, let $class(A_i)$ denote a class s_i of E_i. Each data unit p sent by A_i has a security class $class(p) = class(A_i)$, i.e. $p.CLASS = class(E_i)$.

Suppose that A_i receives a data unit p from A_j. If $class(A_j)(= s_j) \rightarrow class(A_i)(= s_i)$, A_i *accepts* p. Then, the security class of the data ($p.DATA$) carried by p is changed to s_j. If not, E_i rejects p.

4.2 Data Transmission

We present the data transmission procedure to keep the information flow among the entities regal in the cluster. Let C be a roled cluster $\langle E_1, ..., E_n \rangle$. Each entity E_i can transmit a data unit p to the entities in the cluster if the following condition is satisfied.

[Transmission] Let $p.DST$ be a set $\{ E_{i_1}, ..., E_{i_{m_i}} \}$ $(\subseteq C)(n_i \geq 1)$. If $class(E_i) \rightarrow class(E_{i_1}) \cap ... \cap class(E_{i_{m_i}})$, E_i sends p to $E_{i_1}, ..., E_{i_{m_i}}$ in C. □
That is, if some entity E_j has a security class such that not $class(E_i) \rightarrow class(E_j)$, E_i cannot send p in C.
[Example] Suppose that there are some entity E_i and E_j in a group C whose security classes are s_i and s_j, respectivery. That is, there are two security $class(E_i)$ and $class(E_j)$. Suppose that $class(E_i) \rightarrow class(E_j)$. Some entity E_i can send p to E_j. If not, E_i cannot send p to E_j.

E_i can accept a data unit p from E_j if the following condition is satisfied.

[Acceptance] Let $p.DST$ be a set $\{ E_{j_1}, ..., E_{j_{m_j}} \}$. If $class(E_{j_1}) \cup ... \cup class(E_{j_{m_j}}) \rightarrow class(E_i)$, E_i accepts p in C. □
That is, if some destination E_{j_k} of p has a security class such that not $class(E_{j_k}) \rightarrow class(E_i)$, E_i cannot receive p.
[Example] Suppose that there are some entity E_i and E_j in a cluster C whose security classes are s_i and s_j, respectivery. That is, there are two security $class(E_i)$ and $class(E_j)$. Suppose that $class(E_j) \rightarrow class(E_i)$. Some entity E_i can receive p. If not, E_i cannot receive p.

5 Concluding Remarks

In this paper, we have discussed how to control the information flow in the cluster of multiple entities on the basis of the security class. We have presented the data transmission procedure which keeps the information flow *regal* in the cluster.

Reference

[1] Birman, K. P., Schiper, A., and Stephenson, P., "Lightweight Causal and Atomic Group Multicast," *ACM Trans. on Computer Systems*, Vol.9, No.3, 1991, pp.272-314.

[2] Denning, D. E. R., "Cryptography and Data Security," *Addison-Wesley*, 1982.

[3] Kasahara, H., Morita, N., Ito, T., and Imai, K., "ATM Ring Architecture and Its Application to High-Speed and Multi- Media Networks," IPSJ MDP, Vol.91, No.38, 1991, pp.87-94.

[4] Nakamura, A. and Takizawa, M., "Reliable Broadcast Protocol for Selectively Ordering PDUs," *Proc. of the IEEE ICDCS-11* 1991, pp.239-246.

[5] Nakamura, A. and Takizawa, M., "Priority-Based Total and Semi-Total Ordering Broadcast Protocols," *Proc. of the IEEE ICDCS-12*, 1992, pp.178-185.

[6] Takizawa, M. and Mita, H., "Secure Group Communication Protocol for Distributed Systems," *Proc. of the IEEE COMPSAC'93* 1993, pp.159−165.

[7] Sandhu, R. S., "Lattice-Based Access Control Models" *IEEE Computer*, Vol.26, No.11, 1993, pp. 9−19.