7C−7

Information Flow Control in Group Communication with Lattice Model *

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

1 Introduction

In the group communication, multiple entities send messages to the destination entities and can receive the messages sent in the group. [1] discussed how to provide application entities with the atomic and ordered delivery of the messages to the destinations in the group. In addition to supporting the atomic and ordered delivery, it has to be guaranteed that each entity receives messages from only the entities in the group, i.e. *authenticity*, and only the entities in the group send the massages in the group, i.e. *secrecy*. [3] discusses how to support the authenticity and secrecy in the group communication.

One entity would like another entity not to receive the messages sent by it in the group. After receiving message p from entity E_i, if E_j forwards p to another E_k in the group, E_k can receive p from E_i. Even if E_i sends p to E_j but not E_k, E_k receives p from E_j. It is an illegal information flow from E_j to E_k. When considering the secure group communication, the illegal information flow has to be prevented. The *lattice-based information flow model* [2] is discussed to be a model for keeping all the information flows legal.

Each E_i in some group may send messages to another group to which E_i does not belong. Thus, information may be flown from one group to another. In this paper, we would like to discuss such *inter-cluster* information flow and give rules to keep the inter-cluster information flow legal.

In section 2, we present a model of the communication system. In section 3, we present the lattice model of security classes. In section 4, we discuss the data transmission procedure on the basis of the security classes. In section 5, we discuss how to control the information flow among multiple groups.

2 System Model

The communication system is composed of *application*, *system*, and *network* layers. The network layer provides the *reliable* high-speed broadcast communication [1]. The entities at the system layer can communicate with each other by using the network layer to provide the application entities with the secure group communication. Application entity A_i takes the service through system service access point (SAP) S_i supported by system entity E_i. A *cluster* C is a set of the system SAPs S_1, ..., S_n. 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.

In the group communication, each message p sent by E_i is delivered to all the entities in C. [1] discusses a selective group communication where E_i can send each message p to any subset of C, not necessarily all the entities in C. In this paper, we assume that the network layer supports the selective secure group communication. A_1, ..., A_n first reguest the system layer to establish a cluster C among them. C is established by the cooperation of E_1, ..., E_n. Then, A_i can send each message to only and all the destinations in C, i.e. *secrecy*, and can receive messages destined to A_i from only the entities in C, i.e. *authenticity*. [3] discusses how to realize the secure group communication by using the public key system.

In addition to realizing the secrecy and authenticity of the group communication, the information flow among the application entities has to be controlled, i.e. messages not be forwarded to the entities which are not the destinations. In this paper, we would like to discuss how to provide the application entities with the secure information flow by using the reliable broadcast network supported by the network layer.

3 Lattice-Based Model

We would like to present briefly a lattice-based model [2] to deal with the information flow. Let S be a set of security classes. Every entity belongs to one security class. Information in each entity has the security class of the entity. The *can-flow* relation $\rightarrow \subseteq S^2$ is defined as follows. For every pair of security classes s_1 and s_2 in S, information of s_1 can be flown into entities of s_2 iff $s_1 \rightarrow s_2$. For example, suppose that an individual p has a security class s_p and a database D has a security class s_D. If $s_D \rightarrow s_p$, p can obtain the data in D. The information flow model is represented in a lattice $\langle S, \rightarrow, \cup, \cap \rangle$ where $\cup$ is a least upper bound (*lub*) and $\cap$ is a greatest lower bound (*glb*) on $\rightarrow$. For every pair of security classes s_1 and s_2 in S, $s_1 \cup s_2$ is s in S such that $s_1 \rightarrow s$, $s_2 \rightarrow s$, and no s_3 in S such that $s_1 \rightarrow s_3$, $s_2 \rightarrow s_3$, and $s_3 \rightarrow s$. $s_1 \cap s_2$ is defined similarly. Here, $s_1 \succ s_2$ if $s_2 \rightarrow s_1$ but not $s_1 \rightarrow s_2$. s_1 *dominates* s_2 ($s_1 \succeq s_2$) iff $s_1 \succ s_2$ or $s_1 = s_2$. $s_1 \succeq s_2$ means that information of s_1 is more sensitive than s_2.

Suppose that a cluster C supports application entities A_1, ..., A_n. Each A_i is supported by system entity E_i ($i = 1, ..., n$). Each A_i has one security class $class(A_i) \in S$. Each message p sent by A_i has a security class $class(p)$ ($= class(A_i)$). A_i can send message to A_j if $class(A_i) \preceq class(A_j)$. Since $class(p) \preceq class(A_j)$, p can be stored in A_j.

[Example 1] Suppose that there are three application entities A_1, A_2, and A_3 supported by a cluster C, whose security classes are s_1, s_2, and s_1, respectively. Suppose that there is a *can-flow* relation $s_1 \rightarrow s_2$ [Figure 1]. A_1 and A_3 can send messages to A_2, but A_2 can send messages to neither A_1 nor A_3 because $s_1 \preceq s_2$. A_1 and A_3 can communicate with each other because $class(A_1) = class(A_3)$. □

[Definition] Let C be a cluster supporting A_1, ..., A_n. The information flow in C is *legal* iff for every A_i and A_j and for every message p sent by A_i to A_j, $class(A_i) \preceq class(A_j)$. □

---

*束モデルを用いた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

4 Roled Cluster

4.1 Roles

A cluster C is redefined to be a tuple of roles $\langle R_1, ..., R_n \rangle$. Let S be a set of security classes. Let O be a set of primitives which application entities can issue to C, i.e. *send*, *receive*, *open*, *close*, *abort*, and *reset*. Each role R_i is defined to be a pair of a security class $s_i (\in S)$ and a collection $O_i (\subseteq O)$ of primitives which application entities can issue to C, i.e. $R_i = \langle s_i, O_i \rangle$. Let $class(R_i)$ denote s_i and $Op(R_i)$ denote O_i. Suppose that application entities A_1, ..., A_n establish C where each A_i is supported by system entity E_i. Each A_i is referred to as *bound* to C with R_i if C is established by the cooperation of E_1, ..., E_n. It is written as $\langle A_1{:}R_1, ..., A_n{:}R_n \rangle$ named an *instance* of C which denotes a state of C being established. This means that A_i can issue primitives in $Op(R_i)$ to C.

Suppose that A_1 is bound to a cluster C with a role $R_1 = \langle s_1, O_1 \rangle$. If $O_1 = \{receive\}$, A_1 can only receive messages sent in C while A_1 cannot send messages. If $O_1 = \{send, close\}$, A_1 can send messages and close the cluster.

4.2 Constrains

C is represented by a *cluster graph* where each node R_i shows a role R_i and there is a directed edge from R_i to R_j, i.e. $R_i \rightarrow R_j$ if $R_i \preceq R_j$. $R_i \rightarrow R_j$ is *supported* iff $send \in O_i$ and $receive \in O_j$. Even if $s_i \preceq s_j$, unless $R_i \rightarrow R_j$, A_i cannot deliver messages to A_j. R_i and R_j are *linked* (written as R_i — R_j) iff $R_i \rightarrow R_j$ are supported, R_j — R_i, or there is R_k such that R_i — R_k and R_k — R_j. C is *connected* iff for every R_i and R_j, R_i and R_j are linked. If C is not connected, C is partitioned into disjoint subgroups. There is no supported link among any pair of subgroups while every subgroup is connected. This means that there is no way for any two subgroups to communicate with each other. Hence, C cannot be established if C is not connected.

[Example 2] In Figure 1, suppose that $O_1 = \{send\}$, $O_2 = \{receive, send\}$, and $O_3 = \{send\}$. $R_1 \rightarrow R_2$ is supported since $s_1 \preceq s_2$ and $send \in O_1$ and $receive \in O_2$. Neither $R_1 \rightarrow R_3$ nor $R_3 \rightarrow R_2 \rightarrow R_1$ is not supported since neither $receive \in O_1$ nor $receive \in O_3$. Figure 1 shows the information flow and cluster graph. □

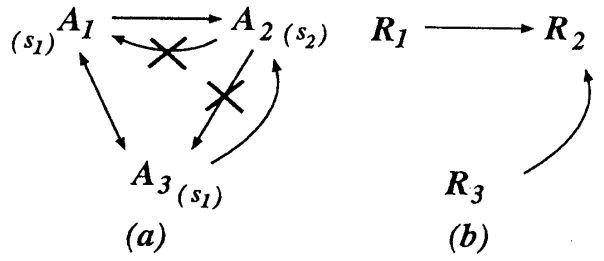

Figure 1: Information flow and cluster graph

Each application entity A_i is modeled as an object which has data structure D_i and a set of operations O_i for manipulating D_i. One property of operation *op* is whether or not *op* derives data from D_i. The other indicates whether or not *op* change D_i. For example, *read* derives the data and *write* changes the data. While each message including an operation which only derives data can be sent to any entity, messages which includes data may not be sent to some entity.

There are types of communication in the cluster. One type is a one-to-many communication like the client-server model. Let S and C_i be roles of server and client ($i = 1, ..., n$). $class(S) \preceq class(C_i)$ if $Op(C_i) = \{receive\}$, i.e. retrieval. Every client can read information in the server. The cluster graph is star-structured. In the other type, each entity sends and receives message equally. Here, every role has the same security class.

5 Inter-Cluster Communication

In some case, entities in a cluster would like to send messages to another cluster. For example, suppose that there are two clusters, *database* cluster R and *teleconference* cluster T. R is composed of redundant database servers. Users in T send update requests of the database to R. Here, information is flown into R from T. We would like to discuss the information flow among clusters.

Suppose that there are two clusters C_1 and C_2 which support secure group communication and legal information flow. Suppose that entity A_i in C_i would like to forward message p to C_j. p has security class s_1 in C_1, and s_2 in C_2. For every pair of security classes s_1 and s_2, information of s_1 can be flown into s_2 iff $s_1 \rightarrow s_2$ according to the definition. There is security class s for s_1 in C_1, and s_2 in C_2, such that $s_1 \succ s \succ s_2$. Information of s can be flown from C_1 to C_2 if $s_1 \succ s \succ s_2$. If not, it cannot be flown. Let C_k denote a roled cluster $\langle A_{ik}{:}R_{ik}, ..., A_{kn_i}{:}R_{kn_k} \rangle$. A_{ij} in C_i can send message p to C_j by the following rule.

[Inter-cluster information flow rule]

(1) $class(p)$ is changed into $s_{i1} \cap ... \cap s_{in_i}$, and

(2) p can be sent to C_j if $class(p) \preceq s_{j1} \cup ... \cup s_{jn_j}$. □

[Example 3] Suppose that there are three clusters, say C_1, C_2, and C_3. Each C_i includes three application entities A_{i1}, A_{i2}, and A_{i3} ($i = 1, 2, 3$). In C_1, A_{11} and A_{13} have security class s_1 and A_{12} has s_2. In C_2, A_{21} and A_{23} have s_3 and A_{23} has s_3. In C_3, A_{31}, A_{32}, and A_{33} have s_1. Here, suppose that $s_1 \preceq s_2 \preceq s_3 \preceq s_4$. Suppose that A_1 would like to send message p to C_2. First, let $class(p)$ be $s_1 \cap s_2$, i.e. s_1. $class(R_{21}) \cup class(R_{22}) \cup class(R_{23}) = s_3 \cup s_1 \cup s_3 = s_4$. Since $s_1 \preceq s_4$, p is sent to C_2. □

In the inter-cluster information flow, each entity in C_i is allowed to send messages to another C_j by using the *lub* of security classes in C_i, and to receive messages by using the *glb* of security classes in C_j. If not, they are rejected.

6 Concluding Remarks

In this paper, we have discussed how to control the information flow in the cluster composed of multiple entities and the inter-cluster information flow on the basis of the security class. We have discussed the mandatory access control on the communication primitives, e.g. *send* and *receive*.

Reference

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

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

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