SECURE BROADCAST COMMUNICATION

Makoto Takizawa
Information and Systems Department
Tokyo Denki University
Ishizaka, Hatoyama, Hiki-gun, Saitama 350-03, Japan
Telephone 0492-96-2911 ext.246
Telefax 0492-96-0501

ABSTRACT

Current information systems are composed of various computer systems interconnected by local area networks (LANs) and radio networks in addition to the conventional wide area networks. The LANs and radio networks provide broadcast communications at the media access control (MAC) layer. That is, every station can receive every protocol data unit (PDU) transmitted by every station in the broadcast communication networks. Although the broadcast communications are required in distributed applications like the distributed database systems, one problem in the broadcast network is how to realize the secure communication. In this paper, we discuss how to provide a secure broadcast communication among multiple entities in the presence of attacks by malicious entities. Our protocol is based on the public key system.

安全放送通信

滝沢　誠
東京電機大学理工学部

従来からの広域通信網に加えて、現在の情報システムは、Ethernet等のローカル・エリア網(LAN)を主要な通信システムとして含んでいる。また、広域の通信システムとして、衛星等を利用した無線通信システムも重要な通信システムとして注目されてきている。これらの通信システムの特徴は、一回のPDUの送信により、複数の実体にPDUを届けれるという放送通信を提供していることである。放送通信は、分散型データベースシステム等の分散システムにおいて複数の実体間での協調動作を行うために重要な機能である。しかし、分散システムの安全性の観点からは、システム内のどの実体も放送されたPDUを受信できるとともに、自由にPDUを放送できるために、従来の一対一通信を基本とした通信システムよりも安全でない。このために、本論文では、放送通信を用いたシステムで、特定の複数の実体間で安全な通信を提供するプロトコルについて述べる。本プロトコルは、各実体が公開鍵方式による暗号機能を提供していることを前提としている。

1. INTRODUCTION

While various communication systems have been widely used on the basis of the open systems architecture like the OSI[OSI], every computer devices like the workstations and mainframes can be easily connected to each other. The open architecture is very desirable from the viewpoint of usability and connectivity. One critical problem in these systems is security. In addition to proper users, malicious users can easily access various resources on the communication networks.

One solution is to encrypt data unit to be transmitted by some key. Public key systems [DEN, IKE] are familiar in the secure communication among two entities. If the secret key is neither hidden nor inferred, the secrecy and authenticity of secure communication are achieved.

At present we have local area networks (LANs) and radio networks in addition to the conventional wide area networks like the OSI network. In the networks, broadcast communications are provided by the media access control (MAC) layer [IEEE]. Current distributed applications like distributed database systems [BER, TAK88a,c] require the broadcast communication among multiple entities in order to cooperate them, i.e. concurrency control, detection of distributed deadlock, and commitment control [BER]. The reliable broadcast communication protocols have been proposed by [TAK87a,b, NAK]. The protocol provides a service by which every entity can receive the same protocol data units (PDUs) in the same order, on the top of the Ethernet MAC service. By using the protocol, the commitment control can be easily implemented [SHIN].

In the broadcast networks, every station can receive every PDU transmitted by every station. Also, every entity can deliver PDUs to all the entities. Thus, the broadcast networks are principally less secure than the conventional one-to-one networks. In this paper, we discuss how to provide a secure broadcast communication among multiple entities in the presence of malicious entities, by using the reliable broadcast service.

In section 2, we present the communication model which is used in this paper. In section 3, a secure cluster concept is defined. In section 4, a procedure to establish a secure cluster is discussed.

2. COMMUNICATION MODEL

A communication system is composed of hierarchical layers. Each (N) layer is composed of (N) entities[Fig.1]. Each entity E_j is given a unique (N) title T_j. Let Title(E_j) denote the title T_j of E_j. E_j takes (N-1) service through an (N-1) SAP S_j. Each (N-1) SAP S_j has a unique address A_j. Let Address(E_j) denote the address A_j of the (N-1) SAP S_j. Here, we assume that each entity E_j has a unique address A_j = Address(E_j).

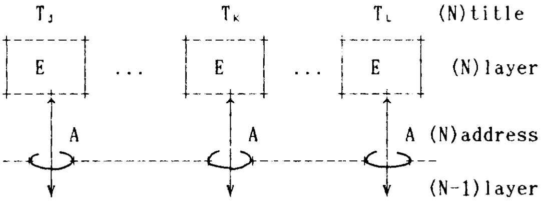


Fig.1 OSI Referential Model

Each (N) protocol data unit (PDU) p includes the following information.

p.SRC = the title of entity which broadcasts p.
p.ADDR= the address of entity which broadcasts p.
p.DEST= the set of titles of entities which expect to receive p.
p.DATA= the data in p.

Here, we make the following assumptions on the (N-1) service.

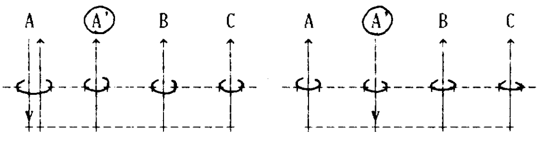


Fig.2 Broadcast Service.

[Assumptions]

(1) If a PDU is sent at an (N-1) SAP, it can arrive correctly at every (N-1) SAP. This implies that entities which broadcast PDUs also receive them.

(2) For every PDU p, every entity cannot write p.ADR in p.

(3) For every PDU p, p.SRC and p.DEST

can be arbitrarily written by the source entity.□

(1) means that the (N-1) service provides reliable broadcast communication among all the (N-1) SAPs [TAK87a,b,88b]. That is, some entity E_k can receive a PDU p although E_k is not the destination of p. Secrecy may be violated. (2) means that the source address is attached automatically to every PDU p and entity cannot update p.ADR. On receipt of a PDU p, every entity E_j understand at which (N-1) SAP p is transmitted. (3) means that every entity E_j writes p.DEST and p.SRC when it broadcasts p. That is, some entity E_m can pretend to be another entity E_j by sending a PDU p and using Title(E_j) in p.SRC. Authenticity may be violated. According to (2), when E_j receives a PDU p, although E_j is not sure whether p is broadcast by an entity whose title is p.SRC, it is assured that p is transmitted at p.ADR by some entity. In order to achieve the secure communication, every entity has to identify the addresses of entities for the titles.

In our communication model, every entity can receive every PDU which was broadcast. On receipt of a PDU p, an entity E_j accepts p if p.DEST includes Title(E_j), otherwise neglects p.

[Definition] An (N-1) cluster C is a tuple $\langle T_1, \ldots, T_n \rangle$ of (N) titles, where T_j is a title of an entity E_j.□

Here, n is a cardinality of the cluster C.

[Definition] An entity E_j is said to be in the cluster C if Title(E_j) = T_j. Otherwise, E_j is said to be outside C.□

Problem is how to securely communicate with each other in the cluster.

3. SECURITY IN CLUSTER

For every PDU p, let p.IDATA denote data which the source entity of p intends to transmit to the destination entities.

[Definition] An entity E_j is said to correctly receive a PDU p iff E_k accepts p and E_k has a transform F_p such that F_p(p.DATA) = p.IDATA.□

Only entity which knows the transform F_p can take intended data carried by p. An enciphering or deciphering is an example of F_p.

Let us consider the secure communication among entities in the cluster.

[Definition] A cluster C is said to be secret iff every PDU broadcast by one entity E_j in C is correctly received by only entities in C including E_j.□

In the secret cluster, any entity outside C cannot correctly receive PDUs broadcast by entities in C.

[Definition] A cluster C is said to be authentic iff only PDU broadcast by one entity E_j in C can be correctly received by an entity in C.□

In the authentic cluster, any entity outside C cannot broadcast to entities in C.

[Definition] A cluster C is said to be secure iff C is both secret and authentic.□

That is, in the secure cluster C, PDUs broadcast by only entity in C are allowed to be received by only and all the entities in C. The PDUs broadcast in C cannot be received by entities outside C. Also, entities outside C cannot broadcast PDUs to the entities in C as if they were in C.

[Definition] A cluster C is said to be established iff all and only the entities in C know a common secret key K. □

The secret key K is said to be a cluster key of C. An algorithm by which a cluster is established among multiple entities is a cluster establishment procedure. After the cluster C is established, it is clear that the cluster is secure because PDUs in C are encrypted by using the secret key K and only entities in C know K. Problem is how to establish the cluster C among multiple entities $E_1, \ldots, E_n$ in the presence of attackers. Attackers are entities which are outside the cluster C but try to join C by pretending entities in C. We formalize the authorization of proper entities in C.

Let EE be a set of all the (N) entities, TT be a set of all the (N) titles, and AA be a set of all the (N-1) addresses. Let CC be a set of possible clusters on EE, i.e. CC $\subseteq 2^{EE}$. We define Name be a function from CC x EE into TT. For a cluster C in CC and an entity E_j in EE, Name(C, E_j) gives a title which E_j uses as its title in C. Name(C, E_j) may not be the same as Title(E_j). In this case, E_j is malicious in the cluster C.

[Definition] Let C be a cluster and E_j be an entity. E_j is said to be proper in C iff Title(E_j) = Name(C, E_j). E_j is said to be malicious in C iff it is not proper in C.□

That is, an entity E_j is proper if it uses its own title in C. Malicious entities pretend to be another proper entities by using a title of another entity. For C, if there exists a malicious entity E_m which pretends to be a proper entity E_j, E_j is said to be pretended and E_m a pretending entity, i.e. for some E_m, Name(C, E_m) = Name(C, E_j). Note that the pretended entity can receive PDUs broadcast by the pretending entity.

For a cluster C, let Dom(C) be a domain of C, which is a set of entities which try to join C, i.e Dom(C) = $\{ E_j \mid$ Name(C, E_j) $\in$ C$\}$. While the cluster is being established, there are two kinds of entities in Dom(C). Ones are active entities, which request the other entities to join C. The others are passive entities, which wait for the request by the active entities. Let Active(C) be a set of active entities in C and Passive(C) be a set of passive entities in C. Since every entity can play only one role, i.e. either passive or active, in C, Dom(C) be a direct union of Active(C) and Passive(C), i.e. Dom(C) = Active(C) + Passive(C).

[Definition] Let C = $\langle T_1, \ldots, T_n \rangle$ be a cluster. A proper domain of C, Pdom(C), is a set of proper entities of C, i.e. Pdom(C) = $\{ E_j \mid$ Title(E_j) = Name(C, E_j) $\in$ C$\}$.□

[Definition] Let C be a cluster. A domain of C, Dom(C), is said to be secure iff Dom(C) = Pdom(C). Dom(C) is said to be establishable iff Dom(C) $\subseteq$ Pdom(C) and Active(C)$\cap$ Pdom(C)$\neq \phi$. Dom(C) is said to be unestablishable iff it is not establishable.□

[Definition] A cluster establishment procedure is complete iff it can establish a secure cluster on only and any establishable domain.□

If there are all the proper entities in the domain Dom(C) and at least one proper entity is active, i.e. Active(C) $\cap$ Pdom(C)$\neq \phi$, the complete cluster establishment procedure can establish a cluster C on Dom(C). For example, if all the active entities are malicious, any complete cluster establishment procedure never establishes the cluster.

4. CLUSTER ESTABLISHMENT PROCEDURE

We present our complete cluster establishment procedure named CEP.

4.1 Reliable Broadcast Network

Our cluster establishment procedure takes advantage of the reliable broadcast communication protocol [TAK87a,b, 88b] provided by the underlying communication network. Our communication system provides a reliable broadcast communication among multiple entities.

[Reliable Broadcast (RB) Service] Every PDU broadcast by an entity can be received by every entity.□

That is, lost and duplicate PDUs never occur in the reliable broadcast service. Fig.3 shows an example of the reliable broadcast service.

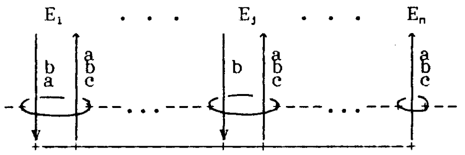


Fig.3 Reliable Broadcast Service

4.2 Basic Protocol

First, we make the following assumptions.

[Assumptions]
(1) Every entity E_j has a secret key S_j

and a public key P_j.

(2) Every entity does not know where entities exist, i.e. addresses of entities except itself.

(3) Every entity knows titles of all the entities.

(4) Every entity knows public keys of all the entities. For each title T, let P_T be a public key of an entity whose title is T.

(5) Every entity is always active. That is, it never fails and always operational.□

Active entities request entities to join the cluster by using their titles but not addresses. In turn, on receipt of the requests, passive entities know the titles of entities which transmit them. Problem is that some malicious entity may use a title of another proper entity. Hence, in the cluster establishment procedure, each entity has to identify the addresses of the proper entities.

For an encryption or decryption Y, we assume that $Y(\langle v_1, \ldots, v_n \rangle) = \langle Y(v_1), \ldots, Y(v_n) \rangle$. Also, we assume that every entity has a bijective function F such that for any tuple $t = \langle v_1, \ldots, v_n \rangle$ of values, F(t) gives a secret key K. Every entity E_k has n variables $t_1, \ldots, t_n$, which are initially set to NULL.

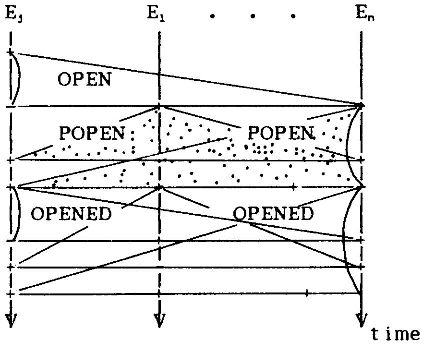


Fig.4 Basic Protocol

[Basic Procedure]

(1) An active entity E_j broadcasts a OPEN PDU p where p.DATA= $S_j(\langle P_1(t_j), \ldots, P_n(t_j) \rangle)$, p.SRC = T_j and p.DST = $\langle T_1, \ldots, T_n \rangle$ to all entities. Here, T_j = Name(C, E_j) (for j = 1,...,n). t_j is a random number generated by E_j or a local clock of E_j.

(2) When E_k receives the OPEN or POPEN PDU p from T_j, where p.DATA = $\langle a_1, \ldots, a_n \rangle$, if Title($E_k$) is included in p.DST, then E_k accepts p and $t_j = S_k(P_j(a_k))$. Set t_k to a number. E_k broadcasts a POPEN PDU p where p.DATA = $S_k(\langle P_1(t_j), \ldots, P_n(t_j) \rangle)$.

(3) On receipt of the OPEN or POPEN PDUs from all the entities, E_k broadcasts an OPENED PDU p where p.DATA = $S_k(\langle P_1(t_j), \ldots, P_n(t_j) \rangle)$.

(4) On receipt of the OPENED PDUs from all the entities, E_k gets a secret cluster key K = $F(\langle t_1, \ldots, t_n \rangle)$. The cluster is established.□

In our basic procedure, a cluster can be established for multiple active entities. In the step (3), it is sure that every entity in C agrees with the same tuple of numbers, $\langle t_1, \ldots, t_n \rangle$. Hence, the entities have the same secure key K = $F(\langle t_1, \ldots, t_n \rangle)$.

4.3 Procedure in the Presence of Attackers

Now, we consider cases that various attackers try to join the cluster. In order to take into account the presence of the attackers, we elaborate the basic procedure. Every entity E_j has the following variables.

t_k = the random number which E_j receives from E_k.

RT_k = a set of tuple $\langle$A, t, type$\rangle$ where A is an address of the entity whose title is T_k, t is its random number, and type is either Active or Passive.

RT = $\langle RT_1, \ldots, RT_n \rangle$

TYPE = Active if E_j ia active, otherwise Passive.

Also, for a tuple $t = \langle v_1, \ldots, v_n \rangle$, let t[j] denote the j-th element v_j of t.

Our secure cluster establishment procedure CEP is shown as follows.

[CEP Procedure]

(1)[Active Entity E_j] E_j sets t_j to a random number, every other variable t_k to NULL (for k= 1,...,n, k≠ j), and Active to TYPE. E_j broadcasts an OPEN PDU p where p.DATA = $S_j(\langle P_1(t_j), \ldots, P_n(t_j) \rangle)$. E_j waits for OPENs or POPENs from all the entities.

(2)[Passive Entity E_k] On receipt of the

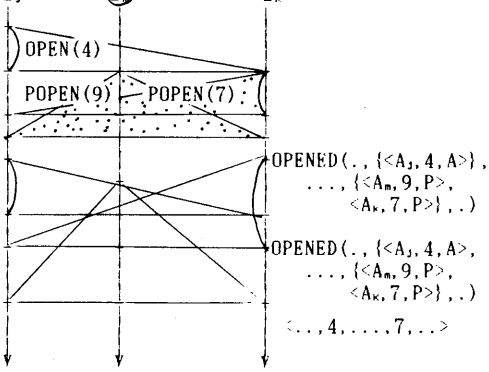
OPEN PDU p where p.SRC = T_j, E_k gets the number u_j = S_j (P_k(p.DATA)[j]). TYPE = Passive. RT_k = RT_k ∪ {<p.ADDR, u_k, Active>}. E_k broadcasts a POPEN PDU p where p.DATA = S_k (<P_1 (t_k), ..., P_n(t_k)>) and p.SRC = T_k. E_k waits for the OPENs or the POPENs from all the entities.

(3)[Passive or Active Entity E_h] On receipt of the OPEN or POPEN PDU p where p.SRC = T_p, if $T_p \neq$ Title(E_h), E_h gets u_p = S_h (P_p (p.DATA)[h]). RT_p = RT_p ∪ {<p.ADDR, u_p, type>} where type = Active if T_p is active, otherwise Passive. If T_p = Title(E_h), then p is neglected.

(4)[E_h receives all OPENs or POPENs] On receipt of the OPENs or POPENs from all the titles in the cluster, E_h waits in a prefixed time. When E_h times out, E_h broadcasts an OPENED PDU p where p.DATA = S_h (<P_1(RT), ..., P_n(RT)>).

(5)[On receipt of OPEN PDU p] Let T be p.SRC and A be p.ADDR, and RRT_j = S_h(P_T (p.DATA)[h])[j] (for j=1,...,n).
(5-1) If RRT_h does not include <Address(E_h), t_h, TYPE>, E_h neglects p and removes every tuple whose address is p.ADDR from RT_T because the entity of p.ADDR is not proper.
(5-2) If RT_T ∩ $RRT_T \neq \phi$, p is neglected like (5-1).
(5-3) For every j, RT_j = RT_j & RRT_j. If some RT_j is empty, E_h aborts.

(6) On receipt of all the OPENs, every RT_j is a singleton in E_h. If all the tuples in RT are passive, E_h aborts the cluster. Otherwise, let K be F(<t_1 ,...,t_n>) where t_j is included in RT_j for j = 1,...,n.□

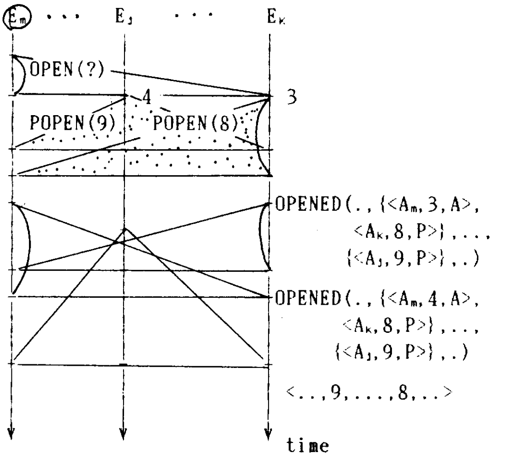


E_J
E_m
E_K
OPEN(4)
POPEN(9)
POPEN(7)
OPENED(.., {<A_J,4,A>}, ..., {<A_m,9,P>, <A_K,7,P>}, ..)
OPENED(.., {<A_J,4,A>, ..., {<A_m,9,P>, <A_K,7,P>}, ..)
<..,4,...,7,..>

Fig.5 Passive Attacker

4.4 Completeness

We show that the procedure is complete. First, suppose that there is some malicious passive entity E_m who pretends to be a proper entity E_j. Some active entity E_a broadcasts the OPEN PDU which carries the random number t_a. Since E_j knows the secret key S_j, it can gets t_a. However, E_m cannot get it. E_j broadcasts its random number t_j, and also E_m broadcasts t_m. The other proper entity E_k receives two number t_j and t_m from the title T_j. It cannot decide which is proper. RT_j includes {<A_j, t_j, Passive>, <A_m, t_m, Passive>}. After receiving the OPEN or POPEN PDUs from all the titles in the cluster, every entity broadcasts the OPENED PDU. E_k receives the OPEN PDU p from E_m. Since RRT_j in p includes t_k, E_k finds that the entity E_m whose address is p.ADR is malicious. Thus, malicious passive entity can never join the cluster.



E_m
E_J
E_K
OPEN(?)
4
3
POPEN(9)
POPEN(8)
OPENED(.., {<A_m,3,A>, <A_K,8,P>}, ..., {<A_J,9,P>}, ..)
OPENED(.., {<A_m,4,A>, <A_K,8,P>}, ..., {<A_J,9,P>}, ..)
<..,9,...,8,..>
time

Fig.6 Active Attacker

Next, let us consider a case that there exists some active attacker. Suppose that an entity E_m is active and pretends to be a proper entity E_k. First, E_m broadcasts an OPEN PDU p. Every entity E_j receives p and gets some number tt_{kj} by deciphering p.DATA. Since E_m may broadcast arbitrary number, each entity may get different number. Also, since E_m does not know the secret key of E_k, E_m does not know the numbers which every entity E_j receives from E_m. E_j broadcasts its number t_j and E_k broadcasts t_k. Thus, E_j receives two numbers tt_{kj} and t_k from the entities of address Address (E_m) and

Address(E_K), respectively. On receipt of the OPEN or POPEN PDUs from all the titles in the cluster, every entity broadcasts the OPENED PDU. If E_m broadcasts the OPENED p, every entity E_j finds that E_m is malicious and E_K is proper. If E_m does not broadcast any PDU, every entity times out and finds that E_m is malicious. In both cases, E_m cannot get the cluster key K since it cannot know any numbers of the entities. Thus, if some active entity E_m is malicious and pretends to be another proper entity E_K, every proper entity E_j including E_K can find which entity, i.e. E_m or E_K, is proper. Here, if all the active entities are malicious, the cluster is not established by the procedure (6). The cluster cannot be established unless there is at least one active proper entity. This means that the cluster is never established maliciously although no entity in the cluster want to join it.

As stated above, it is clear for the following proposition to hold.

[Proposition] The CEP procedure is complete.■

4.5 Performance

The protocol is evaluated in terms of time and PDU complexities. First, we consider the PDU complexity. In our procedure, every entity broadcasts two PDUs, i.e. either OPEN or POPEN, and OPENED PDUs. Let n be the number of proper entities, i.e. the cardinality of the cluster, and m be the number of malicious entities. The maximum number of PDUs broadcast to establish the cluster is 2*(n + m), and the minimum number is 2*n.

Next, let us consider the time complexity. Here, we define a round to be a maximum delay time when a PDU transmitted by one entity is delivered to every entity. At the best case, the cluster can be established by 3 rounds.

We consider the length of the PDU. The length of the OPEN and POPEN PDU is O(n). The length of the OPENED PDU is O((n+m)*n).

5. CONCLUDING REMARKS

In this paper, we present a protocol which creates a secure communication group named a secure cluster among multiple entities by using a broadcast communication service like the Ethernet. In the broadcast network, every entity can receive PDUs broadcast by every entity. Malicious entity can receive PDUs broadcast and broadcast PDUs as another entity. In the presence of these attacks, we have showed a complete protocol named CEP which can establish a secure cluster among multiple entities based on the public key system.

At present, we are implementing the secure cluster on the top of the reliable broadcast communication system [TAK87a,b, NAK].

REFERENCES

[BER] Bernstein, P.A., Hadzilacos, V., and Goodman, N., "Concurrency Control and Recovery in Database Systems," Addison-Wesley, 1987.

[DEN] Denning, D.E., "Cryptography and Data Security," Addison-Wesley, 1983.

[DIFF] Difie, W. and Hellman, M., "New Direction in Cryptography," IEEE Trans. Inf. Theory, Vol. IT-22, No.6, 1976.

[DOD] Defense Communications Agency, "DDN Protocol Handbook," Vol.1 - 3, NIC 50004-50005.

[IEEE] "IEEE Project 802 Local Network Standards-Draft," 1982.

[IKE] Ikeno, S and Koyama, K., "現代暗号理論", 電子通信学会 , 1986.

[ISO] ISO, "Basic Reference Model," IS 7498.No.1-3, 1986.

[MET] Metcalfe, R.M.,"Ethernet: Distributed Packet Switching for Local Computer Networks," CACM, Vol.19, No.7, 1976, pp.395-404.

[NAK] Nakamura, A. and Takizawa, M.,"Totally Ordering Broadcast Protocol on Multi-channel System", IPSJ, DPS, 39-1, 1988.

[OSI] "Data Processing - Open Systems Interconnection - Basic Reference Model," DP7498, 1980.

[SHIN] Shin, S., Miyajima, K., and Takizawa, M., "Commitment Control By Using Reliable Broadcast Communication," IEICE, Data Engineering, 1989.

[TAK87a] Takizawa, M., "Design of Highly Reliable Broadcast Communication Protocol," Proc. of the 11th IEEE COM-

PSAC, Tokyo, 1987, pp.731-740.

[TAK87b] Takizawa, M., "Cluster Control Protocol for Highly Reliable Broadcast Protocol," Proc. of the IFIP Conference on Distributed Processing, Amsterdam, 1987.

[TAK88a] Takizawa, M., "Logic Interface and Communication System for Distributed Database System," Proc. of Joint Scandinavian-Japanese Seminar on Information Modeling and Knowledge Bases, Tampere, Finland, 1988.

[TAK88b] Takizawa, M., "Implementation of Reliable Broadcast Protocol on Unreliable Networks," Proc. of the 3rd Joint Workshop on Computer Networks, Cheju, Korea, 1988, pp.9-18.

[TAK88c] Takizawa, M., "Distributed Fact Base System," to appear in Proc. of the 2nd International Symposium on Interoperable Information Systems, 1988, pp.337-344.