IPSJ Journal Vol. 50 No. 2 737–749 (Feb. 2009)

Regular Paper

Proposal and Performance Evaluation of Hash-based Authentication for P2P Network

Atushi Takeda,[†1,†2] Debasish Chakraborty,[†3]
Gen Kitagata,[†2,†3] Kazuo Hashimoto[†2]
and Norio Shiratori[†2,†3]

Recently, P2P networks have been evolving rapidly. Efficient authentication of P2P network nodes remains a difficult task. As described herein, we propose an authentication method called Hash-based Distributed Authentication Method (HDAM), which realizes a decentralized efficient mutual authentication mechanism for each pair of nodes in a P2P network. It performs distributed management of public keys using Web of Trust and a Distributed Hash Table. The scheme markedly reduces both the memory size requirement and the overhead of communication data sent by the nodes. Simulation results show that HDAM can reduce the required memory size by up to 95%. Furthermore, the results show that HDAM is more scalable than the conventional method: the communication overhead of HDAM is $O(\log p)$.

1. Introduction

In peer-to-peer (P2P) networks, all client nodes mutually communicate directly, using no servers. In fact, P2P networks present many advantages over centralized networks. P2P networks are easy to build, and offer anonymity in communications, etc. Therefore, applications which run in P2P networks are prevalent [1]. However, it is difficult to authenticate nodes in P2P networks, which is an important problem in P2P network operation. Authenticating a node requires validation of a message using an e-signature appended to the message and the sender's public keys [2),3)]. Public Key Infrastructure (PKI) is an existing method of node authentication [4]. It can facilitate effective node authentication based on social trust between the node user and the manager of the certificate authority. In fact, PKI manages authentication information such as public keys assisted by permanent servers called a certificate authority. However, in P2P networks, no node provides permanent services because all nodes alternate between login and logout states. For that reason, managing authentication information with a permanent node such as a certificate authority is difficult in P2P networks.

As described in this paper, we propose a new authentication method called Hash-based Distributed Authentication Method (HDAM), which is an efficient authentication method enabling mutual authentication for all pairs of nodes in the P2P network. The basic idea of HDAM is efficient distributed management of public keys for mutual authentication between two nodes in a P2P network using Web of Trust and Distributed Hash Table (DHT) [5),6)]. In a P2P network, HDAM forms a Web of Trust among all nodes; compared with the conventional method, HDAM markedly reduces the number of public keys managed by a node. Thereby, HDAM markedly reduces the memory requirement by a node. Moreover, HDAM realizes an efficient distributed management of public keys by intelligent deployment of DHT. Consequently, HDAM considerably lowers the overhead of required communication data for participating in a network, or leaving from a network and updating public keys. As described in this paper, the results of computer simulations show that HDAM is more scalable than the conventional method: HDAM can reduce the required memory size by 95%. Moreover, the communication overhead of HDAM is $O(\log p)$, where p is the number of nodes in the P2P network. For that reason, adapting HDAM to a large network is much easier than by the conventional method. Results show that HDAM ensures easy establishment of a secure and large P2P networks. Additionally, it enables creation of many secure decentralized applications such as a conference system and a file sharing system.

The organization of the remainder of this paper is the following. Section 2 presents discussion of existing approaches for authentication. Section 3 describes the proposed method—HDAM—and its details. The advantages of HDAM are demonstrated using computer simulations in Section 4. Finally, in Section 5, we describe the conclusion and future works. The basic HDAM concept and some evaluations are described in an earlier paper [7]. However, some important procedures of HDAM are not explained. Therefore, in this paper, we explain all

†1 Department of Intelligent Information Systems, Tohoku Bunka Gakuen University
†2 Graduate School of Information Sciences, Tohoku University
†3 Research Institute of Electrical Communication, Tohoku University

procedures of HDAM and show evaluation results.

2. Related Works

Authentication methods are classifiable into two categories. An authentication of node identifications confirms whether the node identification is valid. An authentication of user permissions confirms whether the user can use the service. As described in this paper, we specifically examine the first: authentication means validating a message using an e-signature and public keys [3].

Public Key Infrastructure (PKI) is the most widely used method to authenticate nodes [4]. PKI enables an authentication with servers called certificate authority, and authenticates a node using social trust between the node user and the manager of the certificate authority. In a PKI system, users must prepare a certificate authority to authenticate nodes. However, no node provides permanent services in P2P networks because P2P networks are networks in which all nodes alternate between login and logout. Therefore, the application of PKI to a P2P network is difficult.

Pretty Good Privacy (PGP) is an existing authentication method that requires no servers [8]. PGP enables decentralized authentication using Web of Trust, which is a trusting relationship between nodes. In a PGP system, nodes can get a new valid public key from a trusted node. However, accumulating all public keys is difficult because PGP has no information for getting public keys such as routing maps. In PGP systems, nodes require much memory to manage keys and must exchange large amounts of data to exchange keys because an efficient scheme for obtaining public keys is not provided. Information for obtaining public keys is needed for the realization of efficient authentication.

An existing authentication method called self-organized public-key management enables an authentication with no centralized service in an ad-hoc network [9]. In a self-organized public-key management system, all nodes get new public keys automatically from trusted neighbor nodes in an ad-hoc network. However, this method has the same problems as PGP, because nodes have no routing map for obtaining public keys in this method.

Some decentralized authentication methods can accumulate public keys in specific networks systematically such as ad-hoc networks and OSPF networks [10),11]. These methods reduce the required memory size and the related communication overhead. Such reduction is enabled using a routing map of the network and the Web of Trust concept. However, the networks with which we can use these methods are limited because the methods depend on the networks' routing protocol.

The HDAM system proposed in this paper produces a routing map for getting public keys automatically using Web of Trust and DHT. An HDAM system performs an on-demand and efficient distributed authentication in any computer network.

3. Our Proposal: HDAM

3.1 Overview of HDAM

Authentication among all nodes is necessary in a P2P network to support many applications such as conference systems and file sharing systems. However, an efficient authentication method for P2P networks has yet to be realized. Therefore, we propose HDAM.

The number of public keys managed using a node can be reduced if nodes in a P2P network can achieve efficient distributed management of public keys. Additionally, if the public keys were numerically reduced, the memory size and the amount of communication data required by each node could also be reduced. Therefore, efficient distributed management of public keys is important in P2P networks. It is possible to manage public keys in a distributed manner using Web of Trust between nodes in a P2P network. An efficient distributed management of public keys with Web of Trust is possible if information used for obtaining public keys is provided to all nodes. However, in P2P networks, no permanent node such as a certificate authority exists to provide the information because nodes in P2P networks alternate between participation and departure.

Our proposed method, HDAM, enables efficient distributed management of public keys using DHT and safe authentication among all nodes in a P2P network using Web of Trust. In an HDAM system, the information that nodes use for obtaining public keys is provided to all nodes without deployment of a permanent node. In fact, compared to the conventional method, HDAM markedly reduces the memory requirement at each node and the overhead of communication

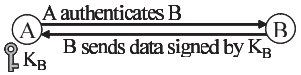

Fig. 1 Authentication.

data at each node. The basic algorithm of HDAM resembles that of an existing method called Chord [5]. However, HDAM and Chord differ in their protocols and distributed management schemes. The objects managed using an HDAM system are public keys. Chord expects that the managed objects are contents such as text, sound, and movies. Therefore, HDAM requires a new protocol and a new distributed management scheme that differ from Chord because the protocol and the distributed management scheme depend on the kinds of managed objects. As described in this paper, we present an authentication protocol with Web of Trust and a distributed management of public keys with DHT. Specifically, we present an authentication procedure with Web of Trust formed DHT. Moreover, we describe procedures performed by nodes in the P2P network for participation, departure, and updating public keys. As described herein, we expect that participating nodes are honest. Therefore, the target of HDAM in this paper is protecting the system from outsider attacks.

3.2 Authentication with Web of Trust

As described herein, a node authentication means validating a message using the e-signature appended to the message and the public key of the node. **Figure 1** portrays the steps in a node authentication process. For two nodes A and B, node A has the public key of node B (K_B), node A can validate messages sent by node B. Therefore, for the discussion herein, the situation in which node A has public key K_B is called "node A authenticates node B". The aggregate of nodes authenticated by node A is designated as $A.trust$.

Figure 2 shows a node authentication method using Web of Trust. The situation portrayed in Fig. 2 (a) is that four nodes A, B, C, and D exist, the status of authentications is $B \in A.trust$, $C \in B.trust$ and $D \in C.trust$, and node A is asked to authenticate node D. In this situation, node A cannot authenticate node D directly because node A has no public key of node D (K_D). The node authentication method with Web of Trust enables that node A gets public key K_D indirectly and can authenticate node D. The node authentication method is

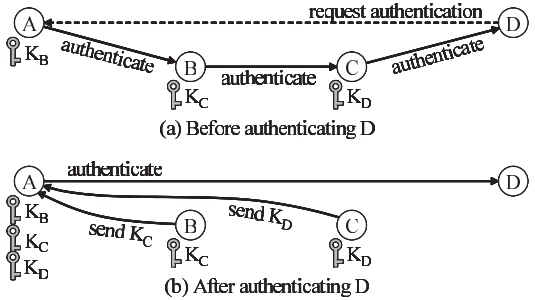

Fig. 2 Authentication with Web of Trust.

the following.

(1) Node A gets K_C from node B:
$\Rightarrow C \in A.trust$.

(2) Node A gets K_D from node C:
$\Rightarrow D \in A.trust$.

Node A gets the public key of node D (K_D) and authenticates node D using K_D. An authentication method as described above, which obtains public keys from trusted nodes indirectly and authenticates new nodes, is called a node authentication with Web of Trust.

3.3 Life cycle of HDAM System

Users of P2P networks can always create HDAM systems anywhere because HDAM systems need no persistent server. An HDAM system starts when a user creates its first node. No specific process is required for creating the HDAM system network. After creating the network, the node can invite other nodes to the created network. Before the node invites the other node, they must authenticate each other without the HDAM system. The HDAM systems are based on the trust assigned through authentication, which is processed without the HDAM system before the invitation. All nodes in the network can invite another trusted node. The HDAM system network is alive as long as it has more than one node in it; the network ceases to exist when all nodes are gone. No specific process is required for finishing the HDAM system network.

3.4 Distributed Management of Public Keys with DHT

Figure 3 presents an example of a distributed management of public keys:

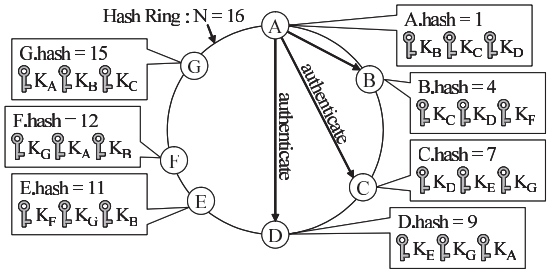

Fig. 3 Distributed management of public keys.

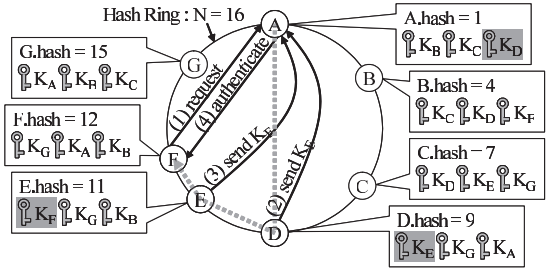

Fig. 4 Authentication procedures.

$i.hash$ is a hash value of node i, K_i is a public key of node i; N is the maximum of the hash value. In an HDAM system, nodes are virtually put on a Hash-Ring based on the hash value, which is derived from the node identification and the one-way hash function. Hash-Ring is a ring in which indexes from 1 to N are put circularly. Node i manages public keys of a forward node that is the nearest node in nodes which are located over $2^k (k = 0, 1, 2, \cdots)$ from node i. In the situation shown in Fig. 3, node A manages three public keys as described below.

- Node A manages a public key of node B, which is the nearest forward node among nodes located more than 2^1 from node A.
- Node A manages a public key of node C, which is the nearest forward node among nodes located more than 2^2 from node A.
- Node A manages a public key of node D, which is the nearest forward node among nodes located more than 2^3 from node A.

In the situation described above, the authentication status is $\{B, C, D\} \subseteq A.trust$. When the number of nodes in the P2P network is p, the number of public keys managed at a node is $O(\log p)$. The maximum number of public keys managed at a node is $\log_2 N$ when the maximum of the hash value is N.

3.5 Authentication Method with Web of Trust formed by DHT

When node n has no public key of node d and when node n is asked to authenticate node d, node n gets the public key of node d by the following steps and authenticates node d.

(1) Node n asks node n^t to send a public key of node d (K_d) to node n. Node n^t is the closest node to node d among nodes that have been authenticated by node n.

(2) If node n^t has public key K_d, node n^t sends public key K_d to node n. Node n authenticates node d using public key K_d.

(3) If node n^t has no public key K_d, node n^t sends a public key of node n' ($K_{n'}$) to node n. Node n' is the closest node to node d among nodes which have been authenticated using node n^t. Node n authenticates node n' using public key $K_{n'}$ and repeats the process from step 1.

Figure 4 depicts an example of the authentication process. In this example, node A tries to authenticate node F using the HDAM authentication method described above.

(1) Node F requests node A to authenticate node F.

(2) Node A has no public key of node F (K_F). Therefore, node A tries to obtain the public key of node F (K_F) from node D because node D is the closest node to node F among the nodes which have been authenticated by node A. However, node D has no K_F. For that reason, node D sends a public key of node E (K_E) in place of K_F. Node A authenticates node E, and the status of authentications is $E \in A.trust$.

(3) Node A repeats the process by trying to get public key K_F from node E. Node E sends public key K_F to node A because node E has public key K_F.

(4) Node A authenticates node F, and the status of authentications is $F \in A.trust$.

Node A obtains public key K_F using the steps described above, and authenticates

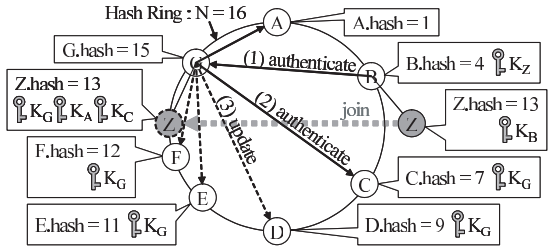

Fig. 5 Participation procedures.

node F. For p nodes in the P2P network, the amount of communication data necessary to authenticate is $O(\log p)$.

3.6 Procedure for Participating

Node n can participate in the P2P network if node n is invited by node g, which has participated in a P2P network. Node n and node g must mutually authenticate without the HDAM system before this invitation. Authentication without the HDAM system is manual authentication such as face to face authentication. Trust in the HDAM system is based on this authentication. The procedure for node n to participate in a P2P network is the following.

(1) Node n gets a public key of node $n.successor$ via node g. Node $n.successor$ is the nearest front node of node n on the Hash-Ring.
(2) Node n builds a Web of Trust. In this process, node n obtains public keys that node n must manage. Node n obtains the public keys via node $n.successor$.
(3) Node $n.successor$ communicates the participation information of node n to all nodes which have the public key of node $n.successor$, because a part of them needs to obtain a public key of node n for rebuilding the Web of Trust.
(4) The nodes which have the public key of node $n.succesor$ rebuild the Web of Trust. In this process, the nodes derive the public keys which they must manage, and a node gets the public key of node n from node $n.successor$ if the node must manage the public key of node n.

Figure 5 portrays an example of a participation procedure of a node in P2P networks. In this situation, node Z participates in a P2P network through the invitation of node B, which is in the P2P network. The example of the participation procedure is the following.

(1) Node Z gets a public key of node G from node B because node G is the nearest front node of node Z on the Hash-Ring.
(2) Node Z gets public keys of nodes G, A, and C, which node Z must authenticate.
(3) Node G communicates the participation information of node Z to nodes E, F, D, and C, which have the public key of node G.
(4) Nodes which have the public key of node G calculate the Web of Trust. In addition, nodes F, E, and D obtain the public key of node Z from node G for rebuilding the Web of Trust.

Nodes participate in a P2P network with the above steps. When the number of nodes in the P2P network is p, the amount of the communication data required by the participation process described above is $O(\log p)$.

3.7 Hash-value Overlap Problem

The hash value of each node is derived from the calculation of the one-way hash function based on node identification. Therefore, the hash values of some nodes participating in a P2P network might be overlapping. In the HDAM system, for overlapping hash values of some nodes, the nodes are arranged in a hierarchical structure. In particular, the first node to participate in the P2P network among nodes which have identical hash values is designated as the parent node. The parent nodes are located on the Hash-Ring, as usual. Nodes with identical hash values to that of the parent node are called "child nodes". The child nodes are not located on the Hash-Ring; they perform all authentication procedures through the parent node. **Figure 6** portrays child nodes' participation procedures.

3.8 Procedure for Leaving

When node n leaves a P2P network, node n sends a departure message to node $n.successor$, which is the nearest front node of node n on the Hash-Ring. Node $n.successor$ communicates the leaving information of node n to nodes which have the public key of node n, because all must obtain the public key of node $n.successor$ for rebuilding the Web of Trust. The nodes which have the public key of node n rebuild the Web of Trust. In this process, the nodes revoke the public key of node n. Then they get the public key of the node $n.successor$ instead of the

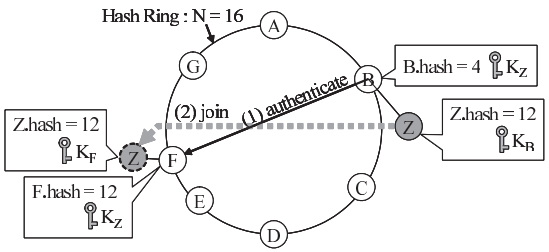

Fig. 6 Participation procedures for child nodes.

public key of node n. Node $n.successor$ detects the event of a node leaving a P2P network without preparation. Node $n.successor$ then informs the other related nodes by sending the necessary messages. For p nodes in the P2P network, the amount of communication data required by the departure procedure is $O(\log p)$.

3.9 Procedure for Updating Public Keys

Periodic updating of public keys is necessary for safe Web of Trust operation. In HDAM systems, each node updates the public key by sending a new public key to nodes that have the old public key. The new public key is sent with an e-signature that can be validated using the old public key. The number of required messages in the updating procedure is the same as the number of public keys managed at a node. Therefore, when the number of nodes in the P2P network is p, the amount of communication data required by the updating procedure is $O(\log p)$.

4. Simulation and Evaluation

4.1 Simulator for P2P Network

To examine HDAM characteristics and evaluate HDAM availability, we developed a simulator of node operations in P2P networks. This system was written in Java; it runs in a Java Runtime Environment. In this simulator, all node operations are implemented in software agents called "node agents". The messages between the node agents simulate all messages sent for participation, departure, updating public keys and sending messages.

Figure 7 depicts the node agents' state transition diagram. Node agents have two statuses: A logout status (S_{out}) means that the node is not in the P2P network; A login status (S_{in}) means that the node is in the P2P network. The probability of changing status S_{out} to status S_{in} is P_{login}, and the probability of changing status S_{in} to status S_{out} is P_{logout}. Moreover, the probability of updating the public key is P_{update}, and the probability of sending a message to a randomly selected node is P_{send}. When a node changes its status to status S_{in}, the node communicates messages according to the procedure described in Section 3.6. All messages contain e-signatures; nodes validate all messages using public keys. All nodes mutually authenticate using the authentication procedure described in Section 3.5.

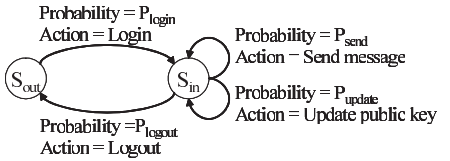

Fig. 7 State transition diagram of node agents.

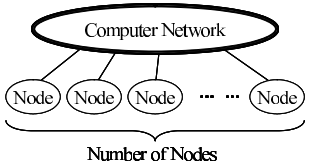

Fig. 8 Network topology assumed in the simulation.

4.2 Simulation Scenario

Figure 8 shows the network topology assumed for this simulation. In this simulation, all nodes are connected by some computer network, such as the Internet; they can communicate mutually. Network failures such as packet loss are not assumed; all communications are executed completely. In the simulation results that follow, the number of nodes represents the number of nodes participating in this computer network.

We evaluated the availability of HDAM in this scenario using the simulator described above. In this simulation, we monitored both the number of public keys managed by a parent node and the number of messages sent by a parent

Table 1 Parameters of agent activities.

scenario	P_{logout}	P_{update}	P_{send}
no. 1	0.45	0.05	0.5
no. 2	0.01	0.01	0.98

node. The number of public keys managed by a node is directly related to the required memory size on a node. The number of messages sent by a node corresponds to the amount of communication data used for authentication.

We considered two simulation scenarios with node agents of two different types. The node agent types are established using the agent activity parameters described above. **Table 1** depicts the configuration parameters of node agents in each scenario, and parameter P_{login} is 1.0 in both scenarios. The node agents in scenario 1 send only a few messages to communication partners. Therefore, they need few public key exchanges for secure communication. The node agent characteristic in scenario 1 is the same as that for applications which join the network temporarily. Usually, these applications are installed into small computers such as PDAs and sensor devices. The applications communicate with a few remote nodes while joining of the P2P network. In scenario 1, a node agent communicates with several nodes during joining the network. Therefore, in this scenario, the performance of an authentication method depends mainly on the participating procedure and leaving procedure. On the other hand, the node agents in scenario 2 send many messages to communicate with their partners. For that reason, numerous public key exchanges are necessary for secure communication. The node agent characteristic in scenario 2 is the same as that for applications that join the network for a long time. Usually, these applications are installed into personal computers; the applications communicate with many remote nodes when joining the P2P network, as in file sharing applications. In scenario 2, a node agent communicates with about 100 nodes when joining the network. Consequently, in this scenario, the performance of public key acquisition described in Section 3.5 is more important than in scenario 1.

4.3 Impact of the Hash-Ring Size

In HDAM systems, all nodes are placed in a Hash-Ring; they mutually communicate using the protocols described in Section 3. The HDAM system performance is therefore dependent on the maximum of the hash value: the Hash-Ring size. **Figure 9** presents the number of public keys managed by each parent node. It shows the memory size that each parent node requires. Each node requires much memory when the maximum hash value is small. However, for higher maximum hash values, the memory required by nodes is smaller.

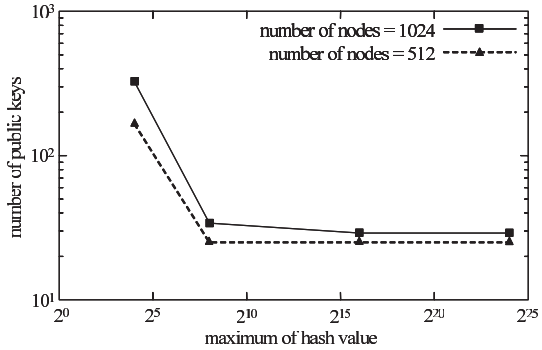

Fig. 9 Number of managed public keys according to the hash-ring size.

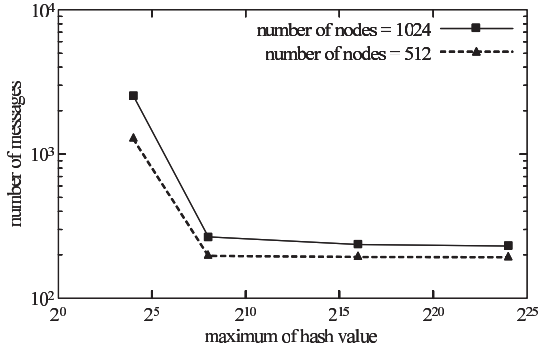

Fig. 10 Overhead of communication data according to the hash-ring size.

Figure 10 portrays the number of messages sent by each parent node. It signifies the communication overhead of HDAM. The communication overhead also gets smaller as the maximum hash value gets larger. Results presented in Figs. 9 and 10 show that the HDAM performance is good when the maximum

hash value is sufficiently large.

4.4 Performance Evaluation

We simulated the conventional method and HDAM in the simulation scenarios described above. To confirm the effectiveness of HDAM, we compare HDAM with a conventional method. In this evaluation, the conventional method authenticates nodes with no centralized servers. The conventional method corresponds to a decentralized authentication method such as PGP and self-organized public-key management [8),9)]. This method performs authentication using Web of Trust, which is not formed by DHT. Therefore, the conventional method must aggregate public keys individually by each node. By the conventional method, a node obtains all public keys when the node participates in the network. Then a node notifies all nodes when the node leaves the network. Similarly, a node sends a new public key to all nodes when the node updates public keys. In the conventional method, nodes mutually authenticate using a public key and an e-signature attached to the message, as in the HDAM system. The conventional method requires large amounts of communication data for participating, leaving, and updating. However, the conventional method needs no communication overhead to send a message. Therefore, the performance of the conventional method in scenario 2 is better than in scenario 1 because the participating and leaving frequencies in scenario 2 are lower than in scenario 1.

4.4.1 Evaluation of Required Memory Size

Figure 11 shows the number of public keys managed by a parent node. Here, the number of public keys represents the required memory size for the authentication system. The solid line in the graph shows the number of public keys in the HDAM system. The dotted line shows the number of public keys in the conventional method. In Fig. 11, the number of public keys managed by parent nodes in the HDAM system is shown to be markedly less than those managed using the conventional method. In particular, when the nodes are 1,024, HDAM can achieve more than 95% reduction in the number of public keys managed by a node, compared with the conventional method. Therefore, HDAM ensures a marked savings in memory requirements at each node compared with the conventional method.

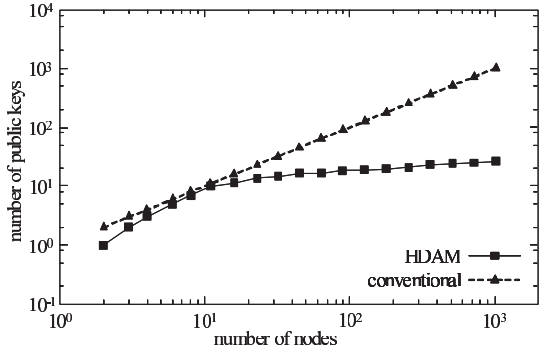

Fig. 11 Number of public keys managed by each node.

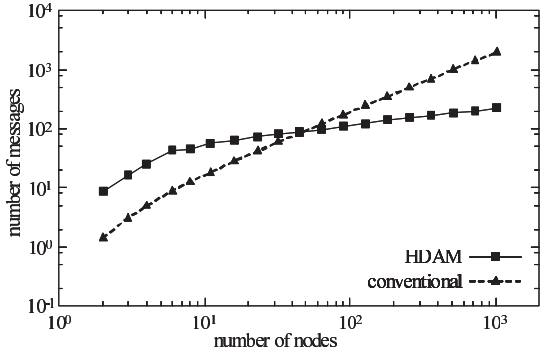

Fig. 12 Communication overhead in scenario 1.

4.4.2 Communication Overhead Evaluation

We evaluate the number of messages sent by a parent node in one step of each scenario described in Section 4.2. The number of messages is the average in more than 200 steps. Node agents perform one action described in Section 4.1 in each step. Furthermore, node agents send some authentication messages in each step. In this evaluation, the number of messages means the communication overhead for node authentication.

Figure 12 shows the number of messages sent by a parent node in scenario 1. The solid line in the graph represents the number of messages in the HDAM

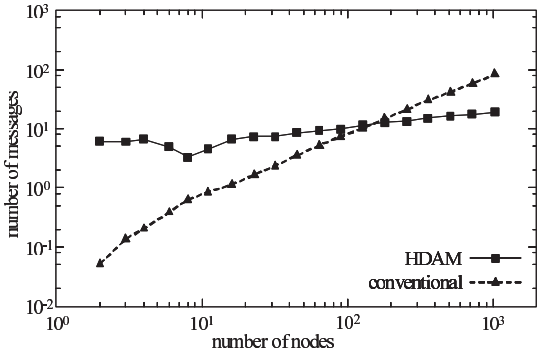

Fig. 13 Communication overhead in scenario 2.

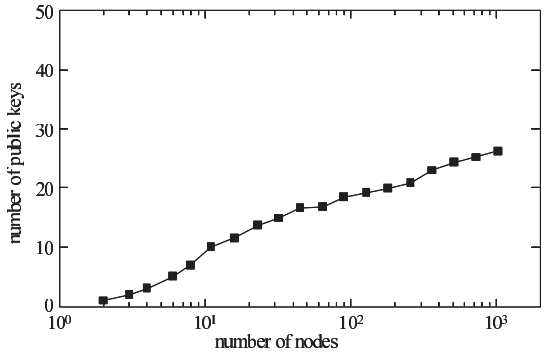

Fig. 14 Number of public keys managed by each node.

system. The dotted line represents the number of messages in the conventional method. As shown in Fig. 12, more messages are sent by a parent node in the HDAM system than in the conventional method when the nodes are fewer than 64 because HDAM needs procedures to build a Web of Trust. However, for more than 64 nodes, the number of messages sent by a parent node in the HDAM system is less than in the conventional method. The gap separating HDAM and the conventional method increases with the number of nodes. For a network of 1,024 nodes, HDAM can achieve 85% reduction in the number of messages sent by a parent node compared to the conventional method.

Figure 13 shows the number of messages sent by a parent node in scenario 2. The solid line in the graph marks the number of messages in the HDAM system. The dotted line marks the number of messages in the conventional method. In this scenario, the advantage of HDAM over the conventional method is less than in scenario 1. Scenario 2 is more unfriendly to HDAM than scenario 1 because P_{send}, which is the probability of sending a message in scenario 2, is higher than in scenario 1. The communication overhead of HDAM for sending a message is larger than the conventional method because the authentication process of HDAM described in Section 3.5 is more complex than in the conventional method. However, the increment of communication overhead of HDAM is less than the conventional method. The participation and departure messages in HDAM are considerably fewer than in the conventional method because the managed public keys in HDAM are markedly fewer than in the conventional method. The communication overhead of HDAM is therefore less than the conventional method when the nodes are sufficiently numerous. Specifically, in scenario 2, which is unfriendly to HDAM, when the number of nodes is 1,024, HDAM can reduce, by more than 60%, the number of messages sent by parent nodes compared with the conventional method.

The advantage of HDAM decreases with P_{send}, which is the probability of sending a message. This feature is observable in scenarios that are intermediate between scenario 1 and scenario 2. However, in all scenarios, the resources required by a HDAM system are less than in the conventional method, when the number of nodes is sufficiently large [7].

4.5 Scalability of HDAM

Figure 14 shows the number of public keys managed at a node in the HDAM system. The data in Fig. 14 are the same as those in Fig. 11, but the scale of the graph in Fig. 14 is semi-logarithmic, which differs from Fig. 11. Figure 14 shows that the number of public keys managed by each node is $O(\log p)$, when p is the number of nodes in the P2P network. Additionally, this result means that the communication data for updating public keys is $O(\log p)$ because the number of messages in a procedure for updating public keys is equal to the number of public keys managed at a node.

Figure 15 portrays the number of hops for the authentication procedure de-

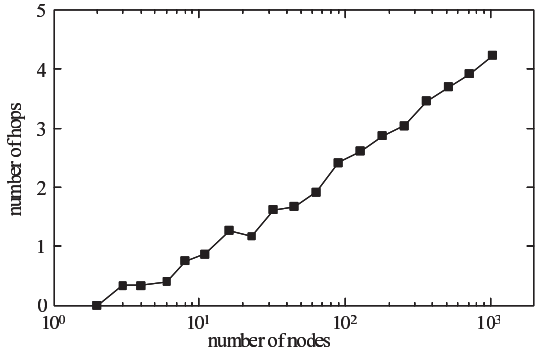

Fig. 15 Number of hops for authentication.

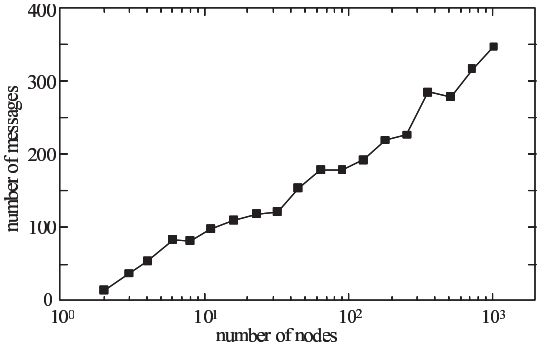

Fig. 16 Number of messages for participation procedure.

scribed in Section 3.5. Figure 15 presents the number of hops for authenticating a node in the HDAM system when p is the number of nodes in the P2P network: $O(\log p)$. The number of messages in an authentication procedure is equal to the number of hops because a node sends a message to each relay node to obtain public keys. Therefore, the result of Fig. 15 dictates that the amount of communication data for authentication is $O(\log p)$. According to Fig. 15, a node sends four messages to authenticate a node when the number of nodes is 1,024.

Figure 16 portrays the number of messages sent by a node in the HDAM system when the node participates in the P2P network; **Fig. 17** presents the

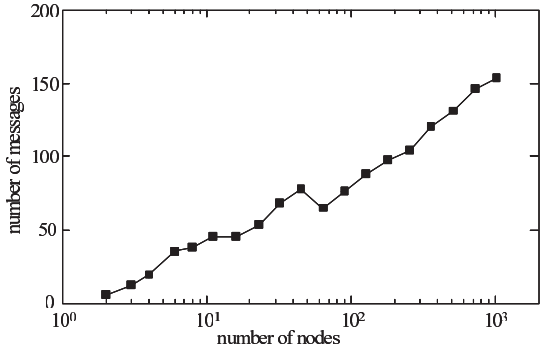

Fig. 17 Number of messages for leaving procedure.

number of messages sent by a node in the HDAM system when the node leaves from the P2P network. The result depicted in Fig. 16 shows that the number of messages used for the participation procedure is $O(\log p)$ when p is the number of nodes in the P2P network. That result means that the amount of communication overhead for participating in the network is $O(\log p)$. The result in Fig. 17 shows that the number of messages for the departure procedure is $O(\log p)$. The result also means that the amount of communication overhead for leaving from the network is $O(\log p)$.

According to Fig. 14, the number of public keys managed at a node in an HDAM system is $O(\log p)$, when p is the number of nodes in the P2P network. Therefore, the memory size required by each node in an HDAM system is $O(\log p)$. The result of Fig. 14 means that the communication overhead for updating public keys is $O(\log p)$; the result presented in Fig. 15 means that the communication overhead for authenticating a node is $O(\log p)$. Moreover, Fig. 16 shows that the communication overhead for participating in the network is $O(\log p)$; Fig. 17 shows that communication overhead for leaving from the network is $O(\log p)$. These results show that the total of the communication overhead of the HDAM system is $O(\log p)$.

4.6 Discussion

Table 2 portrays a comparison of scalability between the proposed method HDAM and the conventional method, which has no servers. When the number

Table 2 Scalability comparison.

	required memory size	communication overhead
conventional method	$O(p)$	$O(p)$
HDAM	$O(\log p)$	$O(\log p)$

p: number of nodes in the P2P network

of nodes is p, the required memory size in HDAM is $O(\log p)$, but the required memory size in the conventional method is $O(p)$. Therefore, when many nodes in the P2P network exist, HDAM enables a drastic reduction of the required memory size. Aside from that, when the number of nodes is p, the communication overhead in HDAM is $O(\log p)$, but the communication overhead in the conventional method is $O(p)$. Therefore, when the P2P network has many nodes, HDAM enables a drastic reduction of the communication overhead. According to the evaluations in Section 4.4, both the memory size requirement by a node and the amount of communication data sent by a node are much less than the conventional method when the number of nodes in the P2P network is sufficiently large. Additionally, the advantage of HDAM increases with the number of nodes. These results show that the scalability of HDAM is better than that provided by the conventional method.

The scalability of an authentication method is important for adapting the authentication method to large networks where many nodes exist. In fact, HDAM reduces the required memory size and communication overhead considerably in large P2P networks; the amount of reduction increases concomitantly with the number of nodes in the P2P network. For example, according to results in Section 4.4, the memory size required by a node in the HDAM system will be less than 0.01% of the memory size required by the conventional method when the number of nodes is a million. Therefore, adapting HDAM to large P2P networks is much easier than the conventional method, and HDAM ensures easy creation of many secure decentralized applications such as a conference system and a file sharing system.

5. Conclusion

Our proposed HDAM, a method for mutual authentication among nodes in P2P networks, enables safe authentication among all nodes in a P2P network using the Web of Trust concept and an efficient distributed management of public keys using DHT. As described herein, HDAM reduces both the memory size needed by a node and the amount of communication data sent by a node. The conventional method requires a bigger memory size and larger communication overhead than HDAM because the conventional method has no efficient mechanism for distributed management of public keys. Therefore, the conventional authentication method cannot run in large P2P networks, where a million nodes might try to communicate at the same time. Our proposed HDAM method can achieve authentication in large P2P networks because of its efficient distributed management mechanism of public keys. Consequently, HDAM is more scalable than conventional methods. Computer simulations have demonstrated that the memory size needed by a node and the communication overhead sent by a node are less than the conventional method when the number of nodes in the P2P network is sufficiently large. For that reason, HDAM is more scalable than the conventional method: adapting HDAM to large networks is much easier than the conventional method. Therefore, HDAM supports the easy establishment of a secure and large P2P network. In addition, HDAM ensures the easy creation of many secure decentralized applications such as conference systems and file sharing systems.

Through our study of the distributed authentication method, we presented the basics of HDAM in this paper. As a future work, we want to establish the details of an HDAM trust model for protecting the system from insider attacks as well as outsider attacks. Our final goal is to realize a secure and large P2P network using HDAM.

Acknowledgments This research was partly funded by the National institute of Information and Communications Technology Japan, under the program of "Research and Development of Dynamic Network Technology", and a Ministry of Education, Culture, Sports, Science, and Technology Grant-in-Aid for Young Scientists, 20700069, 2008.

References

1) Oh, S., Kim, J.-S., Kong, K.-S. and Lee, J.: Closed P2P System for PVR-Based

File Sharing, *IEEE Trans. Consumer Electronics*, Vol.51, No.3, pp.900–907 (2005).
2) Kaliski, B.: RFC 2315: Cryptographic Message Syntax Version 1.5 (1998).
3) Farrell, S. and Housley, R.: RFC 3281: An Internet Attribute Certificate Profile for Authorization (2002).
4) Housley, R., Polk, W., Ford, W. and Solo, D.: RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (2002).
5) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, *IEEE/ACM Trans. Networking*, Vol.11, No.1, pp.17–32 (2003).
6) Zhu, Y. and Hu, Y.: Efficient, Proximity-Aware Load Balancing for DHT-Based P2P Systems, *IEEE Trans. Parallel and Distributed Systems*, Vol.16, No.4, pp.349–361 (2005).
7) Takeda, A., Chakraborty, D., Kitagata, G., Hashimoto, K. and Shiratori, N.: A New Scalable Distributed Authentication for P2P Network and its Performance Evaluation, *The 12th WSEAS International Conference on COMPUTERS* (2008).
8) Garfinkel, S.: *PGP: Pretty Good Privacy*, Oreilly and Associates Inc. (1994).
9) Capkun, S., Buttyan, L. and Hubaux, J.-P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks, *IEEE Trans. Mobile Computing*, Vol.2, No.1, pp.52–64 (2003).
10) Kitada, Y., Watanabe, A., Sasase, I. and Takemori, K.: On demand distributed public key management for wireless ad hoc networks, *Communications, Computers and signal Processing, 2005, PACRIM, 2005 IEEE Pacific Rim Conference on*, pp.454–457 (2005).
11) Goold, J. and Clement, D.M.: Improving Routing Security Using a Decentralized Public Key Distribution Algorithm, *Internet Monitoring and Protection, 2007, ICIMP 2007, Second International Conference on* (2007).

(Received May 19, 2008)
(Accepted November 5, 2008)
(Original version of this article can be found in the Journal of Information Processing Vol.17, pp.59–71.)

Atushi Takeda received both the B.E. degree in Electronic Engineering and M.S. in Information Science from Tohoku University in 2000 and 2002, respectively, and the Ph.D. degree from Tohoku University in 2005. He has been with Tohoku Bunka Gakuen University as an Assistant Professor since 2005. His current research interests include communication network, agent computing and network security. He is a member of IPSJ.

Debasish Chakraborty received his doctoral degree from the Graduate School of Information Science, Tohoku University, Japan in 1999. Presently he is working as a visiting Associate Professor in the Research Institute of Electrical Communication, Tohoku University. He was a TAO research fellow and a NiCT foreign research fellow at Tohoku University Research Center. His main research interests are multicast routing algorithm, QoS, Internet traffic analysis, and wireless and ad hoc networking.

Gen Kitagata is an associate professor of the Research Institute of Electrical Communication of Tohoku University, Japan. He received a doctoral degree from the Graduate School of Information Sciences, Tohoku University in 2002. His research interests include agent-based computing, network middleware design, and symbiotic computing. He is a member of IEICE, IPSJ.

Kazuo Hashimoto received his Ph.D. degree in Information Science from Tohoku University in 2001. He is currently a professor at the Graduate School of Information Sciences of Tohoku University, directing the theoretical and practical study of web communication. He won the Incentive Award from the Japanese Society for Artificial Intelligence in 1999, the Best Paper Award in the field of Artificial Intelligence from the Institute of Electronics, Information and Communication Engineers in 2001, and the achievement award from the Association of Radio Industries and Businesses in 2002. He is a member of IPSJ, IEICE, AAAI, IEEE.

Norio Shiratori is currently a Professor at the Research Institute of Electrical Communication (RIEC), Tohoku University, Japan. Before moving to RIEC in 1993, he was the Professor of Information Engineering at Tohoku University from 1990 to 1993. Prior to that, he served as an Associate Professor and Research Associate at RIEC, Tohoku University, after receiving his Doctoral degree from Tohoku University in 1977. He has also served as the vice Director of RIEC, vice President of IPSJ (Information Processing Society of Japan) and IFIP representative of Japan. He is a fellow of IEEE, IPSJ and IEICE.