A New Scalable Distributed Authentication for P2P Network and its Performance Evaluation

ATUSHI TAKEDA†∗, DEBASHISH CHAKRABORTY‡, GEN KITAGATA†‡,
KAZUO HASHIMOTO† and NORIO SHIRATORI†‡
†Graduate School of Information Science, Tohoku University
‡Research Institute of Electrical Communication, Tohoku University
∗Department of Intelligent Information System, Tohoku Bunka Gakuen University
2-1-1 Katahira, Aoba-ku, Sendai, Miyagi
JAPAN
atushi@shiratori.riec.tohoku.ac.jp, deba@shiratori.riec.tohoku.ac.jp, minatsu@fir.riec.tohoku.ac.jp,
kh@aiet.ecei.tohoku.ac.jp, norio@shiratori.riec.tohoku.ac.jp

Abstract: Recently P2P networks become more and more popular. Though they have many advantages, P2P networks suffer from authentication of nodes. To overcome this problem, a new authentication method called Hash-based Distributed Authentication Method (HDAM) is proposed in this paper. HDAM realizes a decentralized efficient mutual authentication mechanism for each pair of nodes in the P2P network. It performs a distributed management of public keys by using Web of Trust and Distributed Hash Table. Our proposed scheme significantly reduces both the memory size requirement and the overhead of communication data sent by the nodes. Additionally, the results also show that required resource size of HDAM is $O(logn)$ and HDAM is more scalable than the conventional method.

Key–Words: Distributed authentication, Decentralized public key exchange, Peer-to-peer network

1 Introduction

In Peer-to-peer (P2P) networks all client nodes communicate directly with each other without any servers. P2P networks have many advantages over centralized networks. It is easy to build such a network, anonymity in communications etc. Therefore, applications which run in P2P networks are prevalent[1, 2]. However, it is difficult to authenticate nodes in P2P networks, which is an important issue in P2P network operation. Authenticating a node means validating a message by using e-signature appended to the message and public keys of the sender[3, 4]. Public Key Infrastructure (PKI) is a famous existing method of node authentication[5]. PKI can facilitate effective node authentication based on social trust between the node user and the certificate authority-manager. PKI needs the help of permanent servers called certificate authority for managing authentication informations. However, no node can provide permanent services in P2P networks, because in P2P networks, all nodes alternate between login and logout states. Hence, managing authentication information with a permanent node such as certificate authority is difficult in P2P networks.

In this paper, we propose a new authentication method called Hash-based Distributed Authentication Method (HDAM). HDAM is an efficient authentication method that enables mutual authentication for all pairs of nodes in the P2P network. The basic idea of HDAM is efficient distributed management of public keys by using Web of Trust and Distributed Hash Table (DHT). The public key are used for the mutual authentication between two nodes in a P2P network. HDAM forms a Web of Trust among all nodes in a P2P network by using DHT. As a result, HDAM significantly reduces the number of public keys required by a node compared with conventional methods. Thus, HDAM significantly yields a sizable reduction in memory requirement by a node. Moreover, HDAM realizes an efficient distributed management of public keys by intelligent deployment of DHT. Thus, HDAM significantly lowers the overhead of required communication data, sent for authenticating nodes, participating into a network, leaving from a network and updating public keys. In this paper, we observe from the results of computer simulations that HDAM is more scalable than the conventional methods. Specifically, the required memory size of HDAM is $O(logn)$, and communication overhead of HDAM is $O(logn)$. It means that adapting HDAM to a large network is much easier than the conventional methods. HDAM ensures easy establishment of secure and

large P2P networks. In addition, it enables creation of many secure decentralized applications such as a conference system and a file sharing system.

The organization of the rest of this paper is as follows. In section 2, we discuss existing approaches for authentication. In section 3, we present our proposed method HDAM. The advantages of HDAM are shown through computer simulations in section 4. Finally, in section 5, we describe the conclusion and future works.

2 Related Works

Authentication methods can be divided into two main categories. One is an authentication of node identifications, which is to confirm whether the node identification is valid. This is realized by using e-signature and public key. Another is an authentication of user permissions, which is to confirm whether the user can use the service. This is usually realized by using user password[6]. In this paper, we focus on the first, and authenticating means validating a message by using e-signature and public keys[4].

Public Key Infrastructure (PKI) is the most famous authentication method[5]. PKI authenticates a node by using permanent servers called certification authority. The authentication is based on a social trust between the node user and the certificate authority manager. In a PKI system, users have to 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, application of PKI to a P2P network is difficult.

Pretty Good Privacy (PGP) is an existing authentication method which does not need any servers[7]. PGP enables a decentralized authentication by 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, it is difficult to accumulate all public keys, because PGP does not have the information for getting public keys such as routing maps. In PGP systems, nodes require a lot of memory to manage keys and a lot of communication data to exchange keys, because an efficient scheme for obtaining public keys is not provided. The information for obtaining public keys is needed for realization of efficient authentication.

An existing authentication method called self-organized public-key management enables an authentication without any centralized service in an ad-hoc network[8]. In a self-organized public-key management system, all nodes automatically get new public keys from trusted neighbor nodes in an ad-hoc network. However, nodes require a lot of memory to manage keys and a lot of communication data to exchange keys, because nodes do not have a routing map for obtaining public keys.

There are some decentralized authentication methods which can systematically accumulate public keys in specific networks such as ad-hoc networks and OSPF networks[9, 10]. These methods realize a reduction of required memory size as well as communication overhead. The reductions are enabled by using routing map of the network and concept of Web of Trust. However, the kind of networks where we can use these methods is limited, because the methods depend on routing protocol of the networks.

Our proposed, HDAM system, automatically makes a routing map for getting public keys by effectively using Web of Trust and DHT. Therefore, this HDAM system performs an on-demand and efficient distributed authentication in any computer networks.

3 Hash-based Distributed Authentication Method (HDAM)

3.1 Overview of HDAM

Authentication among all nodes in the P2P network is needed by many applications such as conference systems and file sharing systems. However, an efficient authentication method for P2P networks is yet to be realized. Therefore, in this paper, we propose an authentication method that we name as Hash-based Distributed Authentication Method (HDAM).

If nodes in a P2P network can achieve an efficient distributed management of public keys, the number of public keys which is managed by a node is reduced. Additionally, if the number of public keys is reduced, both the memory size and the amount of communication data required by each node are also reduced. Therefore, in P2P networks, an efficient distributed management of public keys is very important. It is possible to manage public keys in a distributed manner by using Web of Trust between each node which participates in a P2P network. If information which the nodes use for obtaining public keys is provided to all nodes, an efficient distributed management of public keys with Web of Trust is possible. However, in P2P networks, there is no permanent node such as certificate authority which provides the information, because all nodes in a P2P network alternate between participation and departure.

Our Proposed method, HDAM, enables efficient distributed management of public keys by using Distributed Hash Table (DHT) and safe authentication among all nodes in P2P network by using Web of Trust. In HDAM system, information which the

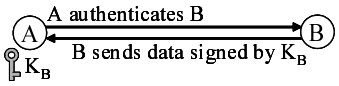

Fig. 1: Authentication

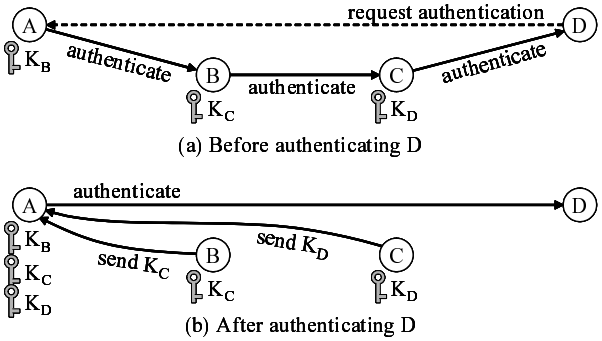

Fig. 2: Authentication with Web of Trust

nodes use for obtaining public keys is provided to all nodes without deploying a permanent node. DHT is often used to manage contents in P2P network[11, 12, 13]. However, HDAM and existing DHT scheme differ in the protocol and the distributed management scheme. The objects managed by a HDAM system is public keys. On the other hand, existing DHT scheme expects that the managed objects are contents such as text, sound and movie. Therefore, HDAM needs a new protocol and a new distributed management scheme. In this paper, we present an authentication with Web of Trust and a distributed management of public keys with DHT. And, we explain an authentication procedure with Web of Trust formed DHT. Moreover, we show that HDAM significantly reduces the memory requirement at each node and the overhead of communication data at each node.

3.2 Authentication with Web of Trust

In this paper, a node authentication means validating a message by using the e-signature appended to the message and the public key of the node. Fig.1 shows the steps in a node authentication process. When two nodes A and B exist, and node A has the public key of node B (K_B), node A can validate messages sent by node B. Therefore, in this paper, the situation that node A has public key K_B is called "node A authenticates node B". And the aggregate of nodes which are authenticated by node A is designated as $A.trust$.

Fig.2 shows a node authentication method with Web of Trust. Fig.2(a) shows the situation where 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

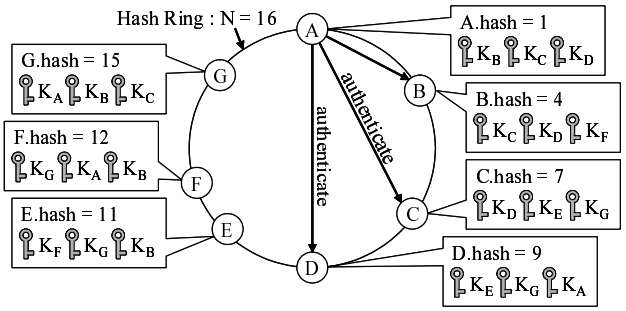

Fig. 3: Distributed management of public keys

node D directly, because node A does not have the public key of node D (K_D). Therefore, node A gets public key K_D indirectly as follows.

1. Node A gets K_C from node B. $\Rightarrow$ $C \in A.trust$.
2. Node A gets K_D from node C. $\Rightarrow$ $D \in A.trust$.

An authentication method as above is called a node authentication with Web of Trust.

3.3 Distributed Management in HDAM

Fig.3 shows an example of a distributed management of public keys. In Fig.3, $i.hash$ is a hash value of node i, K_i is a public key of node i, and N is the maximum of hash value. In HDAM system, nodes are virtually put on a Hash-Ring based on the hash value which is derived from node ID 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 which 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 follows.

- Node A manages a public key of node B which is the nearest forward node in nodes which are located over 2^1 (2^0) from node A.
- Node A manages a public key of node C which is the nearest forward node in nodes which are located over 2^2 from node A.
- Node A manages a public key of node D which is the nearest forward node in nodes which are located over 2^3 from node A.

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

```
// build the authentication path to d
n.authenticate(d)
 begin
  while(!(d ∈ n.trust))
   begin
    n_t := n.closest_trust_node(d)
    n' := n_t.closest_trust_node(d)
    add n' to n.trust
   end
 end

// search for the closest node of d
n.closest_trust_node(d)
 begin
  n' := n
  foreach(n.trust : n_t)
   begin
    if(n_t.hash ∈ (n'.hash, d.hash))
     then
      n' := n_t
     endif
   end
  return n'
 end
```

Fig. 4: Pseudo-code for authentications

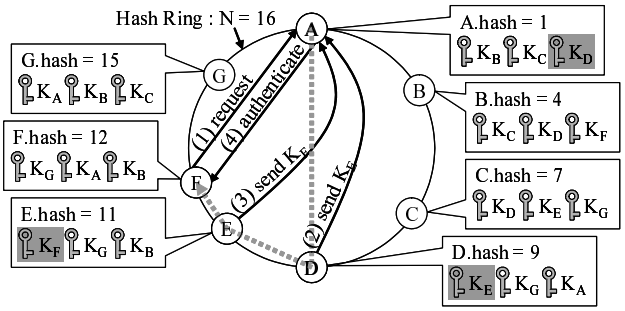

Fig. 5: Authentication procedures

3.4 Authentication Method in HDAM

Fig.4 shows an authentication algorithm in an HDAM system. When node n does not have a public key of node d and is asked to authenticate node d, node n gets the public key of node d by the steps as follows 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 to node d among nodes which 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 by using public key K_d.

3. If node n^t does not have public key K_d, node n^t sends a public key of node n' ($K_{n'}$) to node n. Node n' is the closest to node d among nodes which have been authenticated by node n^t. Node n authenticates node n' by using public key $K_{n'}$, and repeats the process from step 1.

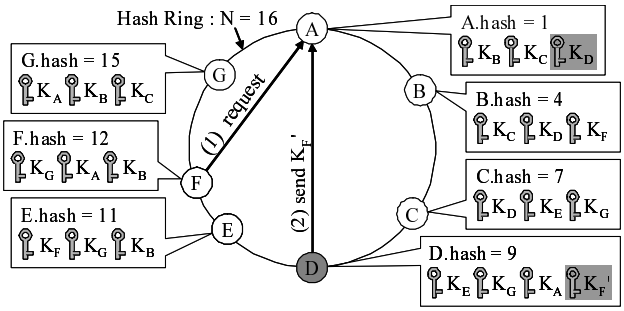

Fig. 6: Problem of insider attack in HDAM system

Fig.5 shows an example of the authentication process. In this example, node A authenticates node F by the HDAM authentication method as above.

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

2. Node A does not have the public key, K_F to node F. Node D is the closest node to node F from node A. So, node A asks node D to send a public key of node F (K_F) to node A. Node D sends a public key of node E (K_E) in place of K_F, because node D does not have K_F. Node A authenticates node E, and the status of authentications is $E \in A.trust$.

3. Node A repeats the process by asking node E to send public key K_F. Node E sends public key K_F to node A.

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

Node A gets public key K_F with the above steps, and node A authenticates node F. When the number of nodes in the P2P network is n, the amount of communication data required to authenticate is $O(log_2 n)$.

3.5 Authentication via multiple nodes

HDAM manages public keys by using Web of Trust. The precondition for Web of Trust is that all nodes are honest. Therefore, HDAM with single Hash-Ring is not resistance to insider attacks from dishonest nodes, Fig.6 shows an example of insider attack in HDAM

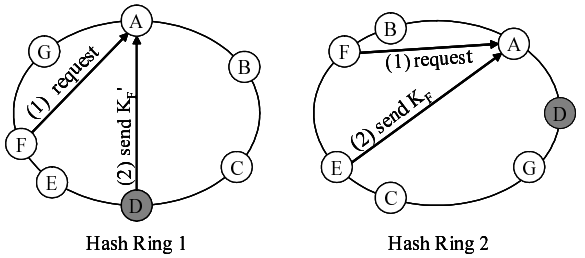

Fig. 7: Authentication via multiple Hash-Ring

system. In this example, node D, which is a dishonest node, sends an invalid public key K_F' to node A. In this situation, node A believes that K_F' sent by dishonest node D is valid, because node A can not confirm if the received public key is invalid. Thus, node A try to authenticate node F by using the invalid public key K_F'. But, the authentication process will be failed, because the valid public key of node F is different from K_F'. In this example, node A can not authenticate node F, and a secure communication between node A and node F is impossible.

The authentication process fails when a node gets an invalid public key from a dishonest node in HDAM system. This is because the node receives a public key from one node only. If a node in HDAM system can get a public key from more than one node, the node can validate the public key by comparing the public keys sent by several nodes. If the public key is same as others, the public key is valid. On the other hand, if a public key is different from others, the public key might be invalid.

HDAM enables the confirmation of public keys by using several Hash-Rings. HDAM system can have more than one Hash-Rings. The position of a node is decided from hash value which is derived from node ID and Hash-Ring number. The hash value is calculated by using one-way hash function such as MD5 and SHA1. Therefore, the positions of a node in each Hash-Ring are different, In HDAM system which has several Hash-Rings, a node can get a public key from several nodes, and the node can confirm the valid public key by comparing public key data received from several nodes. If nodes find the valid public keys, nodes can authenticate other nodes.

Fig.7 shows the example of authentication process in HDAM system. In this example, HDAM system has two Hash-Rings, which are HashRing1 and HashRing2. The deployments of nodes in each Hash-Ring are different. And node D is a dishonest node. When node A wants to get the public key of node F, node A try to get the public key K_F in each Hash-Ring. In this example, node A gets an invalid pubilc key K_F' from node D in HashRing1, because node D is dishonest. And node A gets a valid public key K_F

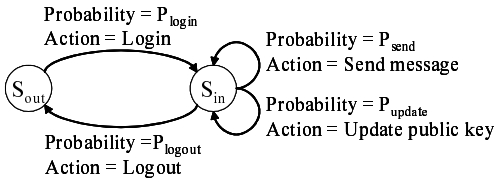

Fig. 8: State transition diagram of node agents

from node E in HashRing2, because node E is honest. In this situation, node A can receive public key K_F' and public key K_F, and node A can detect that one of them is invalid, because public key K_F' and public key K_F are different.

In HDAM system, several Hash-Rings enable that nodes get a public key from several nodes. Thus, HDAM system must manage the several Hash-Rings. Therefore, the required memory size and the amount of communication overhead in HDAM increase with the number of Hash-Rings. When the number of Hash-Rings is m, the required memory size and the amount of communication overhead is $O(m)$. Useally, the number of Hash-Rings is, however, much less than the number of nodes. The number of Hash Rings impacts the scalability of HDAM little. Thus, even if the number of Hash-Rings is more than one, HDAM is scalable enough.

3.6 Life cycle of HDAM system

Users of P2P networks are always able to create HDAM system in anywhere, because HDAM system does not need any persistent servers. A HDAM system starts when a user creates the first node of it. No specific process is required for creating the network of HDAM system. After creating the network, the node can invite ther nodes to the existing network. Before the node invite other nodes, they must authenticate each other without the HDAM system. HDAM system is based on the trust given by the authentication which is processed without HDAM system before the invitation. All nodes in the network can invite another node which is trusted. The network of HDAM system is alive as long as there is more than one nodes in it, and the network ends when all nodes leave from it. No specific process is required to terminate the network of HDAM system. The detail of participation process and departure process of HDAM is described in [14]

4 Simulation and Evaluation

4.1 Simulator for P2P network

In order to examine characteristic of HDAM and evaluate availability of HDAM, we developed a simula-

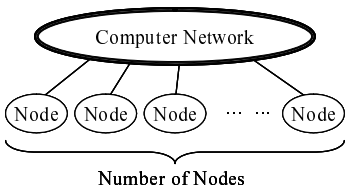

Fig. 9: Network topology assumed in the simulation

tor which simulates operations of nodes in P2P networks. This system is written in Java, and runs on Java Runtime Environment. In this simulator, all operation of nodes are implemented in software agents called "node agent". The messages between the node agents simulates all messages which are sent for participation, departure, updating public keys and sending messages.

Fig.8 is the state transition diagram of node agents. Node agents have two status. One is logout status (S_{out}) which means that the node is leaving the P2P network. The other is login status (S_{in}) which means that the node is joining to 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 of the node whose status is S_{in} is P_{update}, and the probability of sending a message to randomly selected node is P_{send}. All messages contain e-signatures, and all nodes are authenticated by using the authentication procedure described in 3.4.

Fig.9 shows the network topology assumed in this simulator. In this simulator, all nodes are connected by some computer networks like the Internet, and can communicate with each others. Network failures such as packet loss are not assumed, and all communications are executed completely. In simulation results that follow, number of nodes indicates the number of nodes participating in the computer network.

4.2 Tolerance to Insider Attack

Fig.10 shows the success probability of authentication when insider attakers exist in the P2P network. In this figure, the squares show the success probability of authentication in HDAM system that has single Hash-Ring, and the diamonds show the success probability of authenticaion in HDAM system that has three Hash-Rings. This figure shows the relationship between the resistance to insider attacks and the number of Hash-Rings used by HDAM. In this simulation, the number of nodes in the P2P network is 500, and the rate of attackers means the percentage of insider attacker nodes among all nodes in the P2P network. For example, when the rate of attackers is

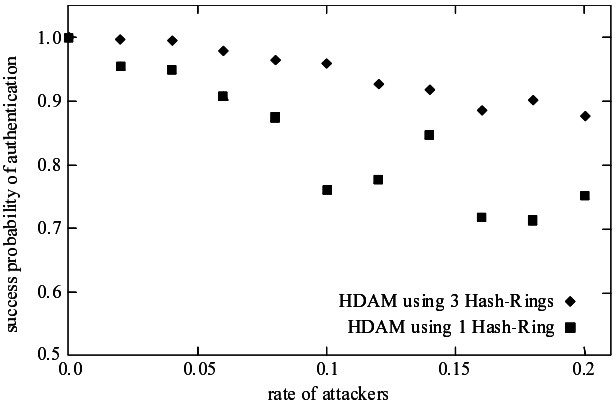

Fig. 10: Success probability of authentication

0.1, the number of insider attacker nodes is 50.

The success probability of authentication in both HDAM systems decreases with the number of insider attackers. But, the success probability in HDAM system that has three Hash-Rings is more than HDAM system that has single Hash-Ring. In HDAM system that has single Hash-Ring, nodes can not validate the public key which is received from other nodes. On the other hand, in HDAM system that has three Hash-Rings, nodes can validate the public key, because nodes can compare the public keys which is sent by three different nodes. For example, when a node receives a public key from three different nodes, the node can validate the public key and authenticate the node which is owner of the public key, even if one of the received public keys is invalid. Therefore, the resistance to insider attacks of HDAM system which has three Hash-Rings is more than HDAM which has single Hash-Ring.

4.3 Performance Evaluation

In order to confirm the effectiveness of HDAM, we compare HDAM with a conventional method that correspond to the best performance parameters. In this evaluation, the conventional method corresponds to a decentralized authentication method such as PGP and self-organized public-key management[8, 7]. The conventional method authenticates nodes without a centralized server. This method performs authentication by using Web of Trust which is not formed by DHT. Therefore, it needs to aggregate public keys individually by each node. In the conventional system, a node aggregate all public keys when the node joins the P2P network, and the node uses them to authenticate each others. Thus, in the conventional method, a node needs to communicate each others for public key exchange when the node joins the network. Additionally, a node needs a memory space to manage public keys. In conventional system, a node does not

scenario	P_{logout}	P_{update}	P_{send}
no.1	0.45	0.05	0.5
no.2	0.25	0.05	0.7
no.3	0.05	0.05	0.9
no.4	0.01	0.01	0.98

Table 1: Parameters of agent activities

however need any communication for public key exchange when the node authenticate others.

We simulated the conventional method and HDAM in the simulation scenarios described above. In this simulation, the maximum of hash value, which is a parameter of HDAM, was set to 2^{24}.

4.3.1 Simulation Scenario

We evaluated the availability of HDAM by using the simulator described above. In this simulation, we monitored both the number of public keys managed by nodes and the number of messages sent by nodes. The number of public keys managed by nodes directly relates to the required memory size on nodes, and the number of messages sent by nodes corresponds to the amount of communication data for the authentication.

We considered four simulation scenarios with four different types of node agents. The types of node agents are established by agent activity parameters described above. Table 1 shows the configuration parameters of node agents in each scenarios, and parameter P_{login} is 1.0 in all scenarios. The node agents in scenario 1 send a few messages to communicate with its partners, so they need small number of public key exchanges for secure communication. The characteristic of node agent in scenario 1 is the same as the applications which join the network for a short time. On the other hand, the node agents in scenario 4 send a lot of messages to communicate with their partners, so they need a lot of public key exchanges for secure communication. The node agent characteristic in scenario 4 is same as the applications which join the network for a long time. Scenario 2 and scenario 3 are intermediate in agent characteristic between scenario 1 and scenario 4.

4.3.2 Evaluation of Required Memory Size

Fig.11 shows the number of public keys managed by a node. Here, the number of public keys means the required memory size for node authentication. The solid line in the graph indicates the number of public keys in HDAM system which has single Hash-Ring, and the dotted line indicates the number of public keys in the conventional method. In Fig.11, it is

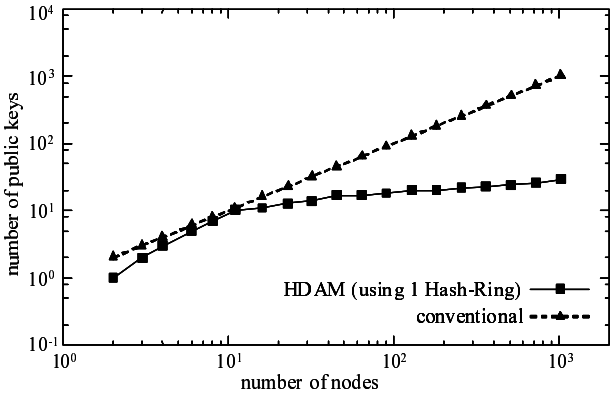

Fig. 11: Number of public keys managed by each node

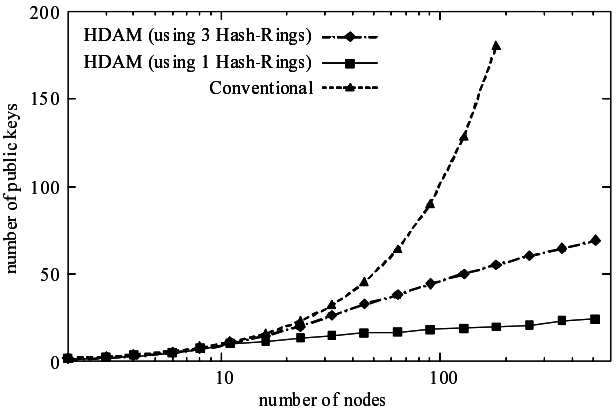

Fig. 12: Number of public keys managed by each node

shown that the number of public keys managed by nodes in HDAM system is significantly less than the conventional method. In particular, when the number of nodes is 1024, HDAM can achieve more than 95% reduction in the number of public keys managed by nodes compared with the conventional method. This means that HDAM ensures a significant savings in memory requirements at each node compared with the conventional method.

Fig.12 also shows the number of public keys managed by a node. As described above, the number of public keys means the required memory size for node authentication. The solid line in the graph indicates the number of publick keys in HDAM system that has single Hash-Ring, and the dashed-dotted line indicates the number of public keys in HDAM system that has three Hash-Rings. The number of public keys in HDAM system that has three Hash-Rings is three times as many as HDAM system that has single Hash-Ring. This means that the memory size required by HDAM system that has three Hash-Rings is three times as much as HDAM system that has single Hash-Ring. However, the scalability of both HDAM

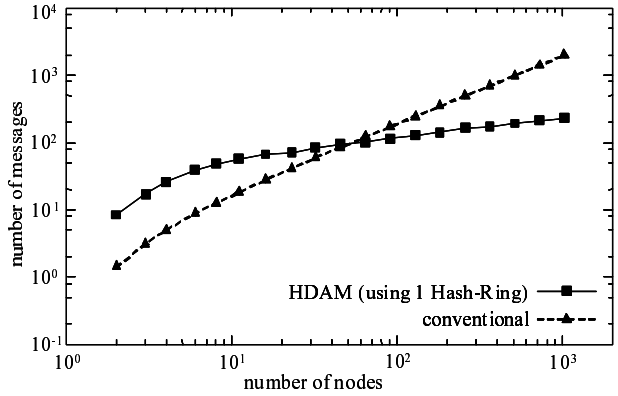

Fig. 13: Communication overhead in scenario 1

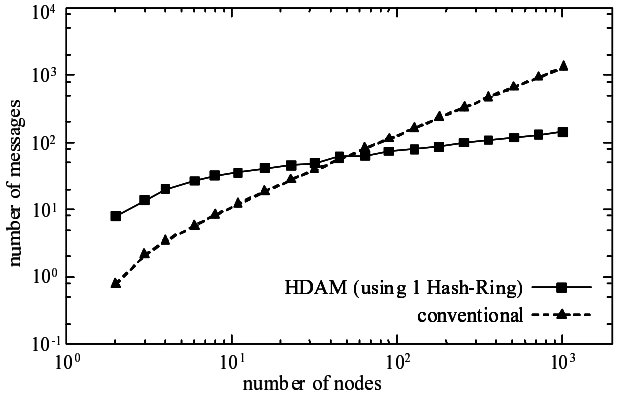

Fig. 14: Communication overhead data in scenario 2

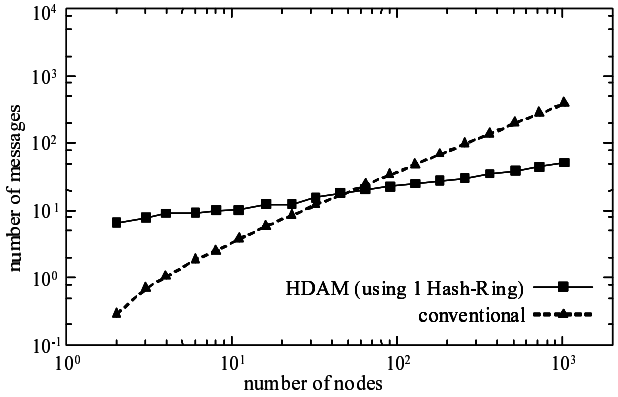

Fig. 15: Communication overhead in scenario 3

systems is better than the conventional method which is indicated by the dotted line in this graph. Thus, the number of public key in HDAM system that has three Hash-Rings is less than the conventional method.

4.3.3 Evaluation of Communication Overhead

We evaluate the number of messages sent by a node in one step of each scenario described in 4.3.1. The number of messages is the average in more than 200

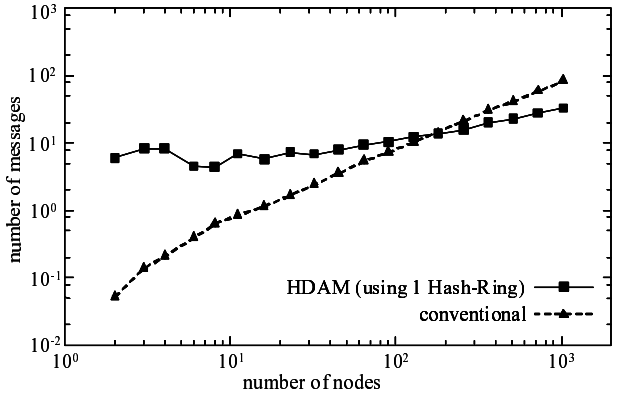

Fig. 16: Communication overhead data in scenario 4

steps. Node agents acts an action shown in 4.1 in each step, and node agents send some authentication messages in each step. In this evaluation, the number of messages means the communication overhead for node authentication.

Fig.13 shows the number of messages sent by a node in scenario 1. The solid line in the graph indicates the number of messages of HDAM which has single Hash-Ring, and the dotted line indicates the number of messages of conventional method. In Fig.13, it is shown that the number of messages sent by a node in the HDAM system is more than the conventional method when the number of nodes is less than 64, because HDAM needs procedures to build the Web of Trust. However, the number of messages sent by a node in the HDAM system is less than the conventional method when the number of nodes is more than 64. And the gap between HDAM and the conventional method increases with the increase in number of nodes. When the number of nodes is 1024, HDAM can achieve 85% reduction in the number of messages sent by nodes compared to the conventional method.

Fig.14, Fig.15 and Fig.16 show the number of messages sent by a node in scenario 2, 3 and 4. In these scenarios, the advantage of HDAM over conventional method is less than scenario 1, because P_{send} which is the probability of sending a message is higher than scenario 1. Specifically, scenario 4 where P_{send} is the highest is the most unfriendly scenario to HDAM in all scenarios. The communication overhead of HDAM in the sending message action is larger than conventional method, because HDAM's authentication process described in 3.4 is more complex than conventional method. However, the increase of communication overhead of HDAM is smaller than the conventional method. The the number of participation and departure messages in HDAM is significantly less than conventional method, because the number of managed public keys in HDAM

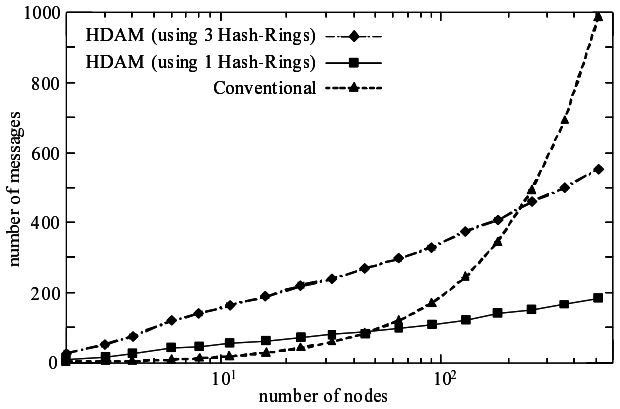

Fig. 17: Communication overhead in scenario 1

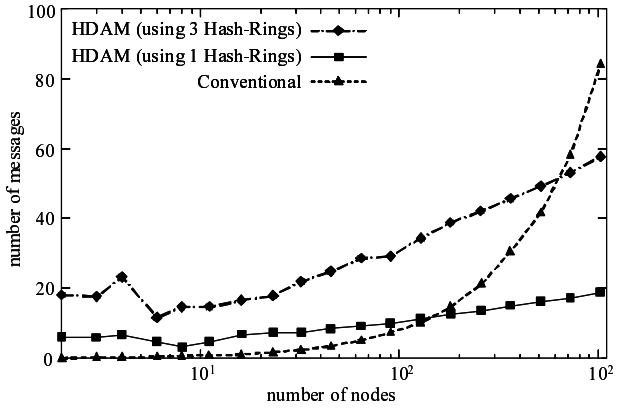

Fig. 18: Communication overhead data in scenario 4

is significantly less than conventional method. The communication overhead of HDAM is therefore less than conventional method when the number of nodes is large enough. Specifically, in scenario 4 which is the most unfriendly scenario to HDAM, when the number of nodes is 1024, HDAM can reduce more than 60% of the number of messages sent by nodes compared with the conventional method.

Fig.17 shows the number of messages sent by a node in scenario 1, and Fig.18 shows the number of messages sent by a node in scenario 4. As described above, the number of messages means the communication overhead for node authentication. The solid line in the graph indicates the number of messages in HDAM system that has single Hash-Ring, and the dashed-dotted line indicates the number messages in HDAM system that has three Hash-Rings. The number of messages in HDAM system that has three Hash-Rings is three times as many as HDAM system that has single Hash-Ring. This means that the communicatino overhead for authentication in HDAM system that has three Hash-Rings is three times as much as HDAM system that has single Hash-Ring. However, the scalability of both HDAM systems is better than the conventional method which is indicated by the dotted line in this graph. Thus, when the number of nodes is large enough, the number of messages in HDAM system that has three Hash-Rings is less than the conventional method.

	required memory size	communication overhead
onventional method	$O(n)$	$O(n)$
HDAM	$O(\log_2 n)$	$O(\log_2 n)$

n : the number of nodes

Table 2: Scalability comparison

4.4 Discussion

Table 2 shows the comparison of scalability between HDAM and the conventional method. When the number of nodes is n, the memory size required by a node in HDAM is $O(log_2 n)$, but the memory size required by a node in the conventional method is $O(n)$. Additionally, the amount of communication data for authentication in HDAM is $O(log_2 n)$, but the amount of communication data for authentication in the conventional method is $O(n)$. Therefore, when there are many nodes in the P2P network, HDAM enables a drastic reduction of the number of messages. This means that the scalability of HDAM is better the conventional method. According to the above evaluations, 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 large enough. Additionally, the advantage of HDAM over the conventional method become more prominent with the increases in number of nodes. These results shows that the scalability of HDAM is better than the conventional method.

5 Conclusion

Our proposed HDAM, which is a mutual authentication method between each node in P2P network, enables safe authentication among all nodes in a P2P network by using Web of Trust and an efficient distributed management of public keys by using DHT. HDAM reduces both the memory size needed by a node and the amount of communication data sent by a node. The scalability of conventional method is less than HDAM, because conventional method has no mechanism for distributed management of public keys. Therefore, conventional authentication methods can not run in huge P2P networks, where a mil-

lion nodes may try to communicate with each other. Whereas, our proposed HDAM method can realize the authentication in huge P2P networks, because of its efficient distributed management mechanism of public keys and thus HDAM is more scalable than conventional methods. Through computer simulations, we have shown that the required memory size and the communication overhead are less than the conventional method when the number of nodes in the P2P network is large enough. It means that HDAM is more scalable than conventional methods, and it means that adapting HDAM to huge networks is much easier than conventional methods. HDAM therefore enables easy establishment of a secure and huge P2P network. Also, HDAM ensures easy creation of many secure decentralized applications such as conference system and file sharing system.

In our study of distributed authentication method we have showed the basics of HDAM in this paper. As a future work, we want to establish the detail of HDAM trust model. Our final goal is to realize a secure and large P2P network by using HDAM.

Acknowledgements: This research was partly funded by National institute of Information and Communications Technology Japan, under the program of "Research and Development of Dynamic Network Technology", Ministry of Internal Affairs and Communications in Japan, SCOPE project(071502003) and the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists, 20700069, 2008.

References:

[1] S. Oh, J.-S. Kim, K.-S. Kong, and J. Lee. Closed p2p system for pvr-based file sharing. *IEEE Transactions on Consumer Electronics*, Vo.51, No.3:900–907, 2005.

[2] J. L. anf Juan R. Diaz, J. M. Jimenez, and M. Esteve. The popularity parameter in unstructured p2p file sharing networks. *WSEAS Transactions on Computer, Issue 6*, 3:2128–2133, 2004.

[3] B. Kaliski. Rfc 2315: Cryptographic message syntax version 1.5, 1998.

[4] S. Farrell and R. Housley. Rfc 3281: An internet attribute certificate profile for authorization, 2002.

[5] R. Housley, W. Polk, W. Ford, and D. Solo. Rfc 3280: Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile, 2002.

[6] A. Tabet, S. Shin, K. Kobara, and H. Imai. On automated analysis of password-based authentication protocols: Csp/fdr model checking and avispa. *WSEAS Transactions on Information Science and Applications, Issue 2*, 6:336–343, 2006.

[7] S. Garfinkel. *PGP : Pretty Good Privacy*. Oreilly and Associates Inc., 1994.

[8] S. Capkun, L. Buttyan, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2, No.1:52–64, 2003.

[9] Y. Kitada, A. Watanabe, I. Sasase, and K. Takemori. On demand distributed public key management for wireless ad hoc networks. *Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on*, pages 454– 457, 2005.

[10] J. Goold and D. M. Clement. Improving routing security using a decentralized public key distribution algorithm. *Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on*, 2007.

[11] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11, No.1:17–32, 2003.

[12] E. Kusmierek, D. H. Du, and J. Beyer. Highly adaptive lookup systems for p2p computing. *WSEAS Transactions on Computer, Issue 6*, 3:1611–1624, 2004.

[13] Y. Zhu and Y. Hu. Efficient, proximity-aware load balancing for dht-based p2p systems. *IEEE Transactions on Parallel and Distributed Systems*, 16, No.4:349–361, 2005.

[14] A. Takeda, K. Hashimoto, G. Kitagata, S. M. S. Zabir, T. Kinoshita, and N. Shiratori. A new authentication method with distributed hash table for p2p network. *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, 2008.