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

Atushi Takeda, Gen Kitagata, Chakaraborty Debasis, Kazuo Hashimoto and Norio Shiratori

In recent years, P2P networks have been evolving at a rapid pace. Authentication of nodes in P2P networks, however, remains a difficult task to conduct efficiently. In this paper, we therefore propose a new authentication method called Hash-based Distributed Authentication Method (HDAM). 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 proposal significantly reduces both the memory size requirement and the overhead of communication data sent by the nodes. Simulation results show that our proposed method, HDAM, can achieve up to a maximum of 95% reduction in the required memory size and a maximum of 80% reduction in the overhead of required communication data compared with the traditional methods.

1. Introduction

Peer-to-peer (P2P) networks are networks in which all client nodes communicate directly with each other without any servers. P2P networks have many advantages over centralized networks in easiness of building the network, anonymity in communications etc. Therefore, applications which run in P2P networks are prevalent. However, it is difficult to authenticate nodes in P2P networks. This problem is important to a P2P network operation. Authenticating a node is validating a message by using e-signature appended to the message and public keys of the sender. There is an existing method of node authentication called PKI (Public Key Infrastructure). PKI can facilitate effective node authentication based on a social trust between the node user and the certificate authority manager. PKI manages authentication informations such as public keys with the help of permanent servers called certificate authority (CA). However, no node provides 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.

Therefore, 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 for the mutual authentication between two nodes in a P2P network by using Web of Trust and Distributed Hash Table (DHT). That is, HDAM forms a Web of Trust among all nodes in a P2P network, and reduces the number of public keys required by a node significantly compared with traditional methods. As such, 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 which is sent when a node participates in a network, leaves from a network and updates public keys. In this paper, we observe from the results of computer simulations that HDAM can enable up to a maximum of 95% reduction in the required memory size and a maximum of 80% reduction in the overhead of required communication data compared with the traditional method when the number of nodes in the P2P network is 500. HDAM ensures easy establishment of secure 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, and describe details of it. 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. Another is an authentication of user permissions, which is to confirm whether the user can use the service. In this paper, we focus on the first, and authenticating means validating a message by using e-signature and public keys.

An existing authentication method called Public Key Infrastructure (PKI) is the most famous method to authenticate nodes. PKI enables an authentication with servers called certificate authority (CA), and authenticates a node by using 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.

An existing authentication method called PGP can authenticate nodes without a server. 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 a routing map for getting public keys. Moreover, nodes require a lot of memory to manage keys and a lot of communication data to exchange keys. An authentication method which provides a routing map for obtaining public keys is needed.

An existing authentication method called self-organized public-key management enables an authentication without any centralized service in an ad-hoc network. In a self-organized public-key management system, all nodes automatically get new public keys from trusted neighbors, because PGP does not have a routing map for obtaining public keys. 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.

An existing authentication method called ad hoc simultaneous nodes search protocol (ASNS) can systematically accumulate public keys by using a routing map of the ad-hoc network. ASNS leverages both a routing map of the ad-hoc network and a concept of Web of Trust. It realizes both a reduction of the required memory size and a reduction of the amount of required communication data. However, deployment of ASNS to networks other than ad-hoc networks is difficult, because ASNS depends on a routing protocol of ad-hoc networks.

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

3. Our Proposal: Hash-based Distributed Authentication Method (HDAM)

3.1 Overview of Our Proposal (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, 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 which are 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 a HDAM system, information which the nodes use for obtaining public keys is provided to all nodes without deploying a permanent node. HDAM significantly reduces the memory requirement at each node and the overhead of communication data at each node that deploy traditional methods. In this paper, we present an authentication with Web of Trust and a distributed management of public keys with DHT. And, we show an authentication procedure with Web of Trust formed DHT. Moreover, we describe procedures which are performed by nodes in the P2P network for participation, departure and updating public keys.
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. The situation shown in fig.2(a) is that four nodes A, B, C and D exist, the status of authentications is B ∈ A.trust, C ∈ B.trust and D ∈ C.trust, and node A is asked to authenticate node D. In this situation, node A cannot authenticate node D directly, because node A does not have the 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 as follows.
1. Node A gets K_C from node B. ⇒ C ∈ A.trust.
2. Node A gets K_D from node C. ⇒ D ∈ A.trust.

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

3.3 Life cycle of HDAM system

Users of P2P networks are always able to create HDAM system in anywhere, because HDAM systems 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 that, the node can invite other nodes to the created network. Before the node invite the other node, they must authenticate each other without the HDAM system. HDAM systems 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 for finishing the network of HDAM system.

3.4 Distributed Management of Public Keys with DHT

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 a max of hash value. In an HDAM system, nodes are virtually put on a Hash-Ring based on the hash value which is derived from 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, ···) 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 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\} ⊆ A.trust. When the max of hash value is N, the max number of public keys managed at a node is \log_2 N.
3.5 Authentication Method with Web of Trust formed by DHT

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 \( n' \) to send a public key of node \( d \) (\( K_d \)) to node \( n \). Node \( n' \) is the closest to node \( d \) among nodes which have been authenticated by node \( n \).

2. If node \( n' \) has public key \( K_d \), node \( n' \) sends public key \( K_d \) to node \( n \). Node \( n \) authenticates node \( d \) by using public key \( K_d \).

3. If node \( n' \) does not have public key \( K_d \), node \( n' \) 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 \( n \). Node \( n \) authenticates node \( n' \) by using public key \( K_{n'} \), and repeats the process from step 1.

Fig. 4 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 authenticates node \( F \). When the max of hash value is \( N \), the amount of communication data required to authenticate is \( O(\log_2 N) \).

3.6 Procedure for Participating

In a situation where node \( n \) participates in a P2P network, the participation method is based on a precondition that nodes \( n \) and \( g \) have mutually authenticated each other where node \( g \) is a node in the P2P network. The procedure for node \( n \) to participate in a P2P network is as follows.

1. Node \( n \) gets a public key of node \( n \)'s successor, which is the nearest front node to node \( n \), from node \( g \).

2. Node \( n \) derives nodes which node \( n \) needs to authenticate, and node \( n \) gets public keys of them from node \( n \)'s successor.

3. Node \( n \) tells nodes in possession of the public key of node \( n \) that rebuilding the Web of Trust is needed.

4. Nodes which receive the message calculate nodes which they need to authenticate. The nodes get public keys of them. If the node needs a public key of node \( n \), the node gets it from node \( n \)'s successor.

Fig. 5 shows 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 participation procedure is as follows.

1. Because node \( G \) is the nearest front node of node \( Z \) on the hash ring, node \( Z \) gets a public key of node \( G \) from node \( B \).

2. Node \( Z \) gets public keys of nodes \( G, A \), and \( C \) which node \( Z \) needs to authenticate.

3. Node \( G \) tells nodes \( E, F, D \) and \( C \) which have the public key of node \( G \) that rebuilding Web of Trust is needed. Nodes which received the message calculate about Web of Trust. Finally, nodes \( F, E \) and \( D \) get the public key of node \( Z \) from node \( G \).

Nodes participate in P2P networks with the above steps. When the max of hash value is \( N \), the amount of the communication data required to participate in a network is \( O(\log_2 N) \).

3.7 Hash-value Overlap Problem

The hash value which each node has is derived
from the calculation of the one-way hash function based on the node identification. Therefore, it is possible that some hash values of nodes participating in a P2P network are overlapping. In HDAM system, in case some hash values of nodes are overlapping, the nodes are arranged in a hierarchical structure. In particular, the first node to participate in the P2P network among nodes which have same hash value is called “parent node”. The parent nodes are located on the hash ring as usual. The nodes whose hash value is same as the parent node are called “child node”. The child nodes are not located on the hash ring, and perform all authentication procedure through the parent node. Fig.6 shows participation procedures of child nodes.

3.8 Procedure for Leaving
When a node leaves a P2P network, the node tells a departure message to the nearest front node. The nearest front node tells nodes which have authenticated the leaving node that rebuilding Web of Trust is needed. Nodes which received the message calculate Web of Trust, and authenticate new node as needed. When a node leaves a P2P network without preparation, the nearest front node first detects the event. It then informs the other related nodes by sending required messages. When the max of hash value is $N$, the amount of communication data required by departure procedure is $O(\log_2 N)$.

3.9 Procedure for Updating Public Keys
Updating public keys periodically is necessary for the safe operation of Web of Trust. In a HDAM system, each node updates the public key by sending a new public key of the node to nodes which have the old public key of the node. When the max of hash value is $N$, the amount of communication data required by the update procedure is $O(\log_2 N)$.

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 simulator which simulates operations of nodes in P2P networks. This system consists of 10000 program lines which were written on 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 and updating public keys.

Fig.7 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 $P_{rand}$. When a node changes its status to status $S_{in}$, the node communicates messages according to the procedure described in 3.6. All messages contain e-signatures, and all nodes are authenticated by using the authentication procedure described in 3.5.

4.2 Simulation Scenario
Fig.8 shows the network topology assumed in this simulation. In this simulation, 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.

We evaluated the availability of HDAM in this scenario by using the simulator described above. In this simulation, we monitored both the number of public keys managed by parent nodes and the number of messages sent by parent 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 performed four simulation scenarios with four different types of node agents. The types of node agents are established by agent activity parameters described above. Fig.9 shows the configuration parameters of node agents in each scenarios, and parameter $P_{login}$ is 1.0 in all of them. The node agents in scenario no.1 send few messages to communication partners, so they need few public keys for secure communication. The node agent characteristic in scenario no.1 is the same as the applications which join the network for a short time. On the other hand, the node agents in scenario no.4 send a lot of messages to communication partners, so they need a lot of public keys for secure communication. The node agent characteristic in scenario no.4 is the same as the applications which join the network for a long time.

4.3 Simulation Results

4.3.1 Impact of Hash-Ring Size

In HDAM system, all nodes are placed in a Hash-Ring and communicate with each others by using the protocols described in 3. The performances of HDAM system are therefore dependent on the Hash-Ring size. Fig.10 shows the number of public keys managed by each parent node. It means the memory size required by each parent node. When Hash-Ring is small, each node needs a lot of memory. But, as the Hash-Ring gets larger, the memory size required by nodes gets smaller. Fig.11 shows the number of messages sent by nodes corresponds to the best performance parameters. In this evaluation, the traditional method authenticates nodes without a centralized server. This method performs an authentication by using Web of Trust which is not formed by DHT. Therefore, the traditional method needs to aggregate public keys individually by each node. The traditional method corresponds a decentralized authentication method such as PGP and self-organized public-key management(1,3). We simulated the traditional method and HDAM in the simulation scenarios which is described above.

Fig.12 shows the relations between the number of
nodes in the P2P network and the number of public keys managed by a parent node. In this graph, $N$ indicates the max of hash value. The solid line in the graph shows the traditional method, and other lines show HDAM. In the fig.12, it is shown that the number of public keys managed by parent nodes in HDAM system is significantly less than the traditional method. In particular, when $N$ is 256 and the number of nodes is 500, HDAM produces more than 95% reduction in the number of public keys managed by nodes compared with the traditional method. This means that HDAM ensures a significant savings in memory requirements at each node compared with the traditional method. The memory savings follows a similar pattern for other values of $N$ as well. This result shows that HDAM is an efficient method.

Fig.13 shows the relations between the number of nodes in the P2P network and the number of messages sent by parent node. The number of messages in fig.13 means the number of messages sent by one parent node at one step. This is the average of the number of messages in more than 200 steps. In this graph, $N$ means the max of hash value. The solid line in the graph shows the traditional method, and other lines show HDAM. In fig.13, it is shown that the number of messages sent by a parent node in the HDAM system is more than the traditional method when the number of nodes is less than 40, because HDAM needs procedures to build the Web of Trust. However, the number of messages sent by a parent node in the HDAM system is less than the traditional method when the number of nodes is more than 40. And the gap between HDAM and the traditional method increases with the number of nodes. In particular, when $N$ is 256 and the number of nodes is 500, HDAM produces more than 80% reduction in the number of messages sent by parent nodes compared with the traditional method. When the number of nodes is $p$, the number of messages for authentication in the traditional method is $O(p)$, but the number of messages in HDAM is $O(\log p)$. Therefore, when there are many nodes in the P2P network, HDAM enables a drastic reduction of the number of messages.

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 traditional method when the number of nodes in the P2P network is more than 40. Additionally, the gap between HDAM and the traditional method increases with the number of nodes. These
results shows that the scalability of HDAM is better than the traditional method.

5. Conclusion

In this paper, we proposed HDAM which is a method for mutual authentication between each node in the P2P network. HDAM 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. Fig. 17 is the scalability comparison between proposal method HDAM and conventional methods. Conventional methods need bigger memory size and larger communication overhead than proposal, because they have no mechanism for distributed management of public keys. Therefore, conventional authentication methods can not run in huge P2P networks such as the network where a million nodes communicate with each other. HDAM which is proposal method realizes the authentication in huge P2P networks, because HDAM has the efficient mechanism for distributed management of public keys and HDAM is more scalable than conventional methods. Through computer simulations, we described that the memory size needed by a node and the overhead of communication data sent by a node are less than the traditional method when the number of nodes in the P2P network is 500. HDAM enables easy establishment of a secure P2P network which is larger than ever. Also, HDAM ensures easy creation of many secure decentralized applications such as a conference system and a file sharing system.

We have been studying about distributed authentication method and showed the basics of HDAM in this paper. In the future, we are going to establish the detail of HDAM trust model. Furthermore, we are going to do P2P network simulations with HDAM and show the HDAM scalability through the simulations. Finally, we will implement the HDAM system and realize a huge secure P2P network by using HDAM.

Acknowledgments This research was partially supported by National institute of Information and Communications of Technology Japan and the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists, 00000000, 2008.

References


(Received 00, 0000) (Accepted 00, 0000)

Atushi Takeda was born in 1977.
Gen Kitagata