

An Algorithm of Constructing Certificates Chain Based on the FriendshipPoint Group for P2P Network

Wang Guoyin¹, Wang Yuhua²

¹ Henan Electric Power Survey & Design Institute, Zhengzhou, China
Email: wanggyzz@163.com

² College of Information Science and Engineering, Henan University of Technology, Zhengzhou, China
Email: yuhua.w@tom.com

Abstract—The authentication and authorization based on CA is not suitable for P2P network because of its centralized control. DHT is efficient to search a required certificate in P2P network, but it is not fit for constructing the certificate chains including many certificates. This paper proposes an algorithm of constructing the certificate chain based on the friendship group of points in p2p network. Compared with DHT, It is more efficient because it adopts the structural overlay network in the small world to construct the authorization and authentication certificate chain.

Index Terms—P2P, SPKI, Authentication, Authorization, Certifications Chain

I. INTRODUCTION

With the success of Music download software *Napster*, P2P has been a hot topic in the recent years. Many applications of P2P, spring up like bamboo shoots after a spring rain. Furthermore, Sun release JXTA, the research and development platform of P2P. In P2P network, the resources are widely distributed in the different physical network segment and security domain, the access controls are confronted with tremendous challenges. Traditional security technology could not solve the conundrum, and CA can not be well applied in P2P network with distributed, dynamic and mutable relationship of trust. If the trust management mechanism based on a single node is employed, it takes much time to search a single certificate, owing to the certificates distributed on each node, so it is more difficult to construct a certificate chain including many certificate [1].

However, if the nodes with the same certificate are constructed to be a point group in a P2P network, coupling degree of nodes will be improved; the number of nodes storing certificates will be decreased and the trust management will be tremendously simplified. SPKI/SDSI put forward a relatively perfect algorithm trust management in distributed environment. Dwaine Clarke has proposed an algorithm constructing a certificate chain of SPKI/SDSI on the assumption that all certificates are stored intensively. Sammer and Ajmani provide an algorithm of how to resolve name on condition that the closure is not required [2]. In the

reference implementation of SPKI/SDSIT, Andrew bring forward a distributed security algorithm against network equipment, which adopts client/sever model and need to make up a network of server[3]. These methods are more convenient and efficient in intensive network, but it is not customized in dynamic and peer to peer network. In P2P network, on account of all certificates distributed storage, it will take much time to search the needed certificates. Distributed Hash Table can efficiently solve this problem in the distributed environment, whose classical methods are ConChord [4] and CAN [5]. Though DHT can obtain fast information search, it will cost much time to construct certificate chains [6].

According to the above analysis, this paper proposes an algorithm of constructing certificate chain of authentication and authorization in P2P network (AAP2P). AAP2P explores the structural P2P network similar to small world, which can be efficient to construct certificate chains. In JXTA, the nodes own joint interest; the resources are managed by groups; it is very ideal to describe and search certificate because of broadcasting and subscribing XML messages. JXTA is convenient for expansion and modification because of being open-source project. So, AAP2P uses JXTA as the development platform.

II. CONSTRUCTING CERTIFICATE CHAINS

The friendship group point for P2P network means that the point group *A* can take place of the point group *B* to execute authentication, authorization, access control and proxy if the point group *B* trust the point group *A*. The core of trust management is that the trust relationships are set among network elements, including the trust relationship of initialization and deduction. The trust relationship of deduction is that the hidden trust relationship will be calculated through constructing a series of certificate chains from the initial trust relationships. In the following section, the algorithm of constructing the certificate chains of authentication and authorization will be described.

A. Authorization

The initial trust relationships can be represented with 5 tuple of authorization certificate in SPKI/SDSI. Suppose that the initial trust relationships consists of

three groups of points A , B and C , and their corresponding PKI are $K_R A$, $K_R B$ and $K_R C$. These three groups of points possess the following certificates:

$$K_R A \rightarrow K_R B \text{ as}$$

$$\text{CertAB} = \langle K_R A, K_R B, A, D, V \rangle \quad (1)$$

$$K_R B \rightarrow K_R C \text{ as}$$

$$\text{CertBC} = \langle K_R B, K_R C, A, D, V \rangle \quad (2)$$

So an authorization certificates chain $K_R A \rightarrow K_R B \rightarrow K_R C$ is created. In the authorization transfer description of SPKI, authorization can be transferred continually if the certificate mark D is "1"; authorization cannot be transferred continually if the certificate mark D is "0". In P2P, a set of nodes authorization $ASet$ is added into each node. If the node possesses the authorization transferred from other nodes, it will not transfer authorization. Six Degrees of Separation shows that a point group will be reached in finite steps. The function *InsertCert* could be called when an authorization certificate is added into the present authorization relationship. *InsertCert* is shown in Fig.1.

Theorem 1 *InsertCert* is a closure of operation. All nodes can obtain corresponding authorization through *InsertCert*.

Proof: Suppose that FS is the trust node set deduced by *InsertCert*. $K_R A$ is the original issuer of authorization certificate, and $K_R Y$ is the trust node of $K_R A$ and not in FS . So a trust path from $K_R A$ to $K_R Y$ should exist: $K_R A \rightarrow K_R M \cdots K_R Y$. According to *InsertCert*, the friendship point group of the point group A should be trusted. $K_R M$ is not the element of FS , so M is not the friendship point group of A . Therefore, $K_R Y$ is not trustful, and it is contradiction with the supposition.

Theorem 2 *InsertCert* cannot lead to a deadlock and authorization certificate cannot be transferred infinitely.

Proof: Suppose another authorization certificate $CertCA$

$$K_R B \rightarrow K_R C \text{ as}$$

$$\text{CertCA} = \langle K_R C, K_R A, A, D, V \rangle \quad (3)$$

According to the equation 1), (2) and (3), a ring $K_R A \rightarrow K_R B \rightarrow K_R C \rightarrow K_R A$ will be obtained. since $K_R A$ owns the certificate $CertCA$, it will not execute *InsertCert* to avoid the deadlock of the ring trust relationship. ■

In addition, the number of steps of *InsertCert* operating is proportional to the length of authorization certificates, which is the length from the original issuer to the object point group.

Corollary 1 The max running time of *InsertCert* is the length of the longest authorization certificate chain, marked. *MaxLenFriPath*

B. Authentication

All users must firstly join in father point group, then it can join in sub-group of points. Nodes will possess all

```

InsertCert( Cert cert )
  if(cert.D is true and cert.
    A is not in ASet )
    friendSet ← cert.S'friends
    ASet ← ASet ∪ cert.A
    for each p ∈ friendSet
      newCert ← <I,p, cert.A∩
        Self.A,D, cert.V∩Self.V>
      InsertCert( newCert )
    else return

```

Figure1. Process of adding authorization certificate

authentications of father group and sub-group. A user must pass all authentications of all point groups from the root node to the object node. A new point group must broadcast its advertisement in its group and the root group, and all nodes can receive its advertisement.

A user executes *Insert* if it want to join in a point group. Fig. 2 shows the operation of insert itself into a point group.

```

Bool AuthenCert (String path, Cert cert)
  if ( path is not empty )
    firstGroup ← GetFirst(path)
    if( cert.A ∈ ASet of firstGroup )
      return true
      AuthenCert( Remainder(path), cert)
    else
      return false

String getFirst(String path)
  return the first name of the path
String remainder(String path)
  return path-getFirst(path)

```

Figure 2. Process of node joining the point Group

GetFirst(path) can catch the first point group of the authentication path. *Remainder(path)* gets the other points group of this path. If the authentication \Group A\Group B...\Group N, *GetFirst(path)* will return Group A, *Remainder(path)* will return \Group B...\Group N.

III. STRUCTURE OF AAP2P

In JXTA, the point group not only includes nodes but also some point groups, so the nesting tree structure will be reached. Two nodes can communicate only if they are in the same and have built the channel link. If there are n nodes in the groups and each two nodes have pipe, there are $n(n-1)/2$ channels in a point group, which leads to waste network resource. AAP2P will solve efficiently it. Fig.3 shows the network structure, which includes the following important elements:

– Head node: to build the link between the father point

group and the sub-point group, at the same time as the member of the father point group and the sub-point group;

- Successor node: the nodes' ID is greater than the

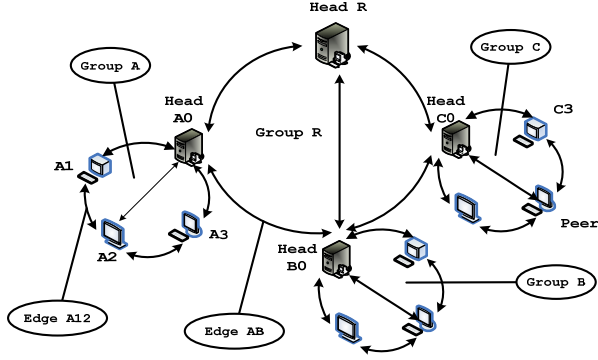


Figure 3. AAP2P network Structure. The nodes in the point group sort descending order according to the nodes' ID, A1、A2、A3 and the node A0 in turn. A2 and A3 are the successor nodes of A1. A1 and A2 are the predecessor nodes A3. A3 is the immediate successor of A2. A1 is the immediate predecessor of A2. A3 and A1 are the adjacent nodes of A2. The edge A12 is the short edge connection for A1 and A2. The edge AB is the long edge connection for the point group A and the point group B.

present node's ID;

- Predecessor node: the nodes' ID is smaller than the present node's ID;
- Immediate successor: the node with the smallest ID among the successor nodes;
- Immediate Predecessor: the node with the biggest ID among the predecessor nodes;
- adjacent node: Immediate successor and Immediate Predecessor;
- Short edge connection: the channel link between two different nodes in the same point group;
- Long edge connection: the channel link to connect the head nodes, but it will be the short edge connection in the father point group.

AAP2P can improve the coupling degree by building the short edge connection between different nodes. According to the nodes' ID, the two-way channel can be build between the nodes, and the node with the biggest ID and the node with the smallest ID will have a two-way channel. The head node is the bridge not only between the father point group and the sun-point group, but also between the different point groups. In JXTA, its core services provide the basic interface for the development of AAP2P. Discovery service provides the essential support for the broadcasting and search resources. In AAP2P, Extended discovery service can broadcast the certificates and the advertisement of point group in the current point group and the root point group; the revised access service can be used to check whether the certificates meet the requirement of resource access; Updated authentication services create certificate through membership services of the point group to check whether the received news comes from the legal members in the point group; the redesigned channel services can

create security and safe channel connection for the different nodes in the point group.

InsertCert will be executed when a new authorization certificate is added to a node in a point group. The mark *I* and *S* of this authorization certificate will be updated with the name of the point group and the name of the friend point group. The point group will pass the certificate to the friendship point group. The validity of the updated certificate will be checked by the access service and authentication service. Suppose that the initial trust relationship is that the point group A trust the point group B and the point group B trust the point group C. When the authentication certificate in the equation (1) *CertAB* is inserted into the initial trust relationship, the head node *B0* of the point group B checks the validity of the certificate *CertAB*. The head node *B0* executes *InsertCert* to save the authentication of certificate *CertAB*, uses the certificate *CertBC* in the equation (2) to update *CertAB*, and transfer the certificate *CertBC* to the head node *C0* of the point group C along the long edge *BC*. *C0* will check the validity of *CertBC* and save the authentication of *CertBC*.

AuthenCert means that all membership services of all point groups from the root point group to the object point group check the validity of the certificate one by one. If all authentications are successful, the node becomes a valid member of the object point group and the object point group issues a member certificate of it. In Fig.3, if C3 want to access the resource of A1, Group R\Group A is the essential authentication certificate chain. The head node of the root point group R executes *InsertCert* to verify whether C3 can meet the requirement of R. If C3 can meet the requirement of R, the head node R transfer the requirement of C3 to the head node A0 in the point group A. A0 verifies whether C3 can meet the requirement of accessing the resource of A1. *InsertCert* and *AuthenCert* can be extended as the service of JXTA, ensures the same services for all nodes in order to be convenient to the role transition.

IV. STRUCTURE OF AAP2P

A. Robustness

Each node uses its ID and pipe as the input parameter of Hash function to produce the corresponding pipe ID [7, 8]. Each two nodes send and receive the messages. Each node sends a ping message to adjacent nodes in the regular time to make sure whether the adjacent nodes is online. If the node receives no ACK message in some time, it considers that the adjacent nodes has been fault or broken the link and the other nodes can reconstruct the two-way pipe ring.

When a node adds into a point group, it can obtain the list of nodes, accesses adjacent nodes and build new pipe with the adjacent nodes. At the same time, it builds a two-way pipe with the head node, receives the copy certificate of it. Therefore, the failure of a node cannot affect the store and search of certificates.

If a head node fails, its successor node will replace it to become a new head node. Since the new head node can obtain the other nodes, it can calculate the pipe ID set by each node and build the two-way pipe. The new head node owns the authentication certificate that the membership service of the father point group issues, so it is a valid node in the father point group. In the father point group, it can be a new node and build the pipe with the adjacent node and the head node of the father point group. This new head node can get the copy certificate about the father point group. Therefore, the new head node can replace the wrong head node, and the system can be recovered.

Suppose that there are N nodes in a point group, each node provides services in the probability P_p and $1 - P_p$ is the probability that it cannot provides. The point group cannot provides service is that $(1 - P_p)^N$, then in the proper status, the probability is the following:

$$P_s = 1 - (1 - P_p)^N \quad (4)$$

Fig.4 shows the probability that a point group works well, in the different nodes of the point group and the different service probability of the point group. The more nodes, the bigger probability can the system provide service in. If there are 250 nodes in a point group and each node provides service in the probability 1%, a point group will provide service in the probability 90%. If each node provides service in the probability 2%, a point group will provide service in the probability 100% approximately. If there are 100 nodes in a point group and each node provides service in the probability 5%, a point group will provide service in the probability 100% approximately. From the recovery algorithms of the system, it can be seen that the system owns the property of self-organization and self-recovery and doesn't falls apart because of the failure of the head node.

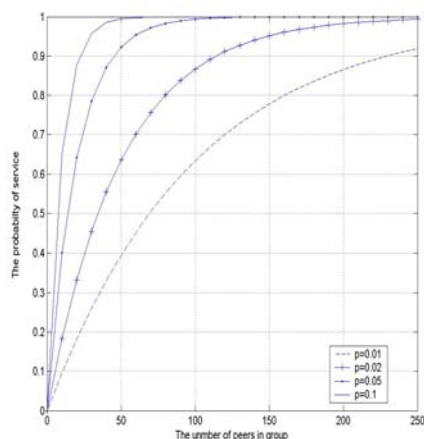


Figure 4. Probability distribution of P_s in N nodes and P_p of a point group. P_s is the probability in which that the point group provide service. N is the number of nodes in a group point, P_p is the probability in which that a node provides

A. Efficiency

AAP2P employs the hierarchical network architecture, the certificates and their copies are stored

into the member nodes. It is convenient to maintain the consistency of certificate copies since the certificate store is refined into the scope of a point group and the copies of certificate is reduce d. The hierarchical architecture of AAP2P improves the degree of coupling of different nodes. The nodes in the different point group communicate through the long edges and the nodes in the same point group communicate through the short edges. The communication efficiency is improved because many communications of nodes in the system are the communications of short edge. The communications of long edge is in favor of reducing the pipe between nodes and avoiding $O(N^2)$ pipe. *InsertCert* will be executed when a node transfers the authentication certificates. Its executing time is decided by *MaxLenFriPath* and the steps are $\log_2(\text{peersNum})$ (PeersNum is the number of nodes in the point group) [9, 10]. If a user join into a point group, AAP2P need to execute *LenAuthPath* times *Authentic*, is the number of the point group from the root point group to the object point. In contrast, the authentication algorithm based on DHT need sto search $\text{LenAuthPath} \times \log_2(\text{peersNum})$ times .

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China under Grant No. 60373087.

REFERENCES

- [1] Li N, Winsborough W H, and Mitchell J C. Distribute credential chain discovery in trust management [C], in Proc. 8th ACM CCS, 2001
- [2] Sameer, Ajmani. How to resolve SDSI Names Without Closure. <http://pmg.Lcs.mit.edu/~ajmani/papers/sdsi-algos.pdf>, 2004-2
- [3] Andrew Mayway. An Implementation of a Secure Web Client using SPKI/SDSI Certificates.Masters's thesis. Electrical Engineering & Computer Science Dept. MIT. June 2000.
- [4] Sammer Ajmani, Dwaine E. Clarke, Chuang-Hue Moh, Steven Richman. ConChord: Cooperative SDSI Certificate Storage and Name Resolution. MIT Laboratory for Computer Science, March 2002
- [5] Ratnasamy, S., et al. A Scalable Content-Addressable Network. in ACM SIGCOMM. 2001. San Diego, CA, USA
- [6] Geng Xiuhua, Han Zhen, Jin Li. Distributed SPKI/SDSI2.0 Certificate Chain Search Algorithm[J], Computer Research and development, 2008, 45(7):1133-1141.
- [7] C. Ellison, B. Frantz, and et.al B. Lampson, "RFC 2693: SPKI Certificate Theory," Sept. 1999.
- [8] Andrew Mayway. An Implementation of a Secure Web Client using SPKI/SDSI Certificates.Masters's thesis. Electrical Engineering & Computer Science Dept. MIT. June 2000.
- [9] Plaxton, C., Rajaraman, R., and Richa, A. Accessing nearby copies of replicated objects in a distributed environment. In Proceedings of the ACM SPAA, June 1997.
- [10] Leonard D, Yao Z, Wang X, Loguinov D. On Static and Dynamic Partitioning Behavior of Large-Scale P2P Networks [C]. Networking, IEEE/ACM Transactions onVolume 16(6), 2008:1475 - 1488