Construction Methods of Closed Users Group using Multi-party Protocol

MASAO TANABE[1], KEITA SUGIYAMA[2], MASAKI AIDA[1]
[1]The Graduate School of System Design
Tokyo Metropolitan University
6-6 Asahigaoka, Hino-shi, Tokyo 191-0065
JAPAN
tanabe@computer.org, maida@sd.tmu.ac.jp
[2]Nomura Research Institute, Ltd.
Marunouchi Kitaguchi Building, 1-6-5 Marunouchi,
Chiyoda-ku, Tokyo 100-0005
JAPAN

Abstract: - In recent years, the Internet has become the most important infrastructure in the information society. Until now, a client-server system such as e-mail or WWW that is offered by a fixed service provider has played an important role. However, in such client-server systems, it is impossible to continue to provide services when the servers are halted by some failure. For this reason, a pure Peer-to-Peer (P2P) network, in which a user also acts as a provider, has lately attracted considerable attention as a new commercial infrastructure, because it enables its users to provide a service without a dedicated server. However, it is difficult to provide security on a pure P2P network because each terminal on the network has the same privilege. In this paper, we study a construction method of a Closed Users Group (CUG) without any administration methods which can control user access to certain applications on a pure P2P network. In our method, we assume that each member of a CUG has a certification which is issued by a certification authority on a public key infrastructure (PKI) that enables members to communicate safely. We realize this PKI on a P2P network using a multi-party protocol. In addition, we study the reliability issues involved in this method and propose some methods of issuing a certification to a new member. Finally, we discuss strong points and applications of one of these methods.

Key-Words: - Closed Users Group, Multi-party protocol, Certification authority, P2P network, Public key infrastructure, I2P Network

1 Introduction

On the Internet, many applications such as e-mail or WWW are provided by client-server systems that rely on central server management. This client-server system has the advantages of easily constructing systems and managing its users and services. On the other hand, it has the disadvantages of having a single point of failure. All services provided by a server will become unavailable if the server stops. For these reasons, a pure Peer-to-Peer (P2P) network in which there is no central dedicated server and clients communicate directly with each other has attracted considerable attention. A pure P2P network has the advantages of being highly scalable and having a high fault tolerance. On the other hand, it has the disadvantages of being difficult to manage its users and services and of providing complicated functions. A pure P2Pnetwork has been used to exchange and share files. Since it has a high fault tolerance, it is expected to be useful for a commercial infrastructure. However, in order to utilize a pure P2P network as a commercial infrastructure, it is necessary to provide security function. Since it is difficult to manage users and services and provide high grade functions, very few pure P2P networks have been utilized as a commercial infrastructure so far. On the contrary, we can utilize pure P2P networks as a commercial infrastructure if we could resolve security issues of them.

For example, as a study on security issues in a pure P2P network, realization of a public key infrastructure (PKI) on a pure P2P network using a multi-party protocol was proposed in [1].

By developing this study, we study a construction method of a Closed Users Group (CUG) without any administration methods which can be accessed by only admitted users to one application which considers simple security function on a pure P2P network in this paper. We introduce a PKI in order to certify users to each other. Moreover, we realize this PKI on a pure P2P network using a multi-party protocol. Then we study the reliability issues on this method and we propose some methods of issuing a certification to a new member. Finally, we discuss strong points and applications of one of these methods.

The remainder of this paper is organized as follows. In Section 2, we discuss how to construct a CUG on a pure P2P network, and explain a construction method of a CUG using a multi-party protocol. In Section 3, we study reliability issues on this method and propose some methods of issuing a certification to a new member. In Section 4, we discuss strong points and applications of one of proposed methods. Finally, Section 5 concludes this paper.

2 Construction CUG on pure P2P network using multi-party protocol

In this section, we discuss how to construct a CUG on a pure P2P network and then explain a construction method of a CUG using a multi-party protocol.

2.1 Construction CUG on pure P2P network

A Public Key Infrastructure (PKI) is a certification system using a cipher (Fig.1). In this figure, a CA is a certification authority and both A and B are users in this PKI. When User A requests its certification from the CA, the CA issues it which is a text encrypted by the CA secret key. Then user A transmits its certification to user B, whom it would like to communicate with. User B confirms user A's certification by using the CA's public key of user A. If user B would like to communicate with user A, it encrypts its message using the public key of user A. If a PKI will be constructed on a pure P2P network, it will be possible to construct a CUG by assuming that user who has its certification belongs to the CUG.

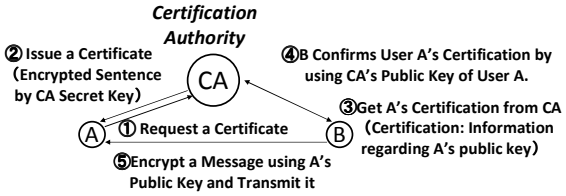

Fig. 1 Public Key Infrastructure (PKI)

In the case of constructing a certification authority (CA) on a pure P2P network, it is desirable to manage it in a distributed manner by its members. However, it is dangerous that a specific user who is assigned an administration role knows the security information stored in the CA. For this reason, we utilize a multi-party protocol [2] to construct the CA on a pure P2P network instead of assigning an administration role to a specific user. This multi-party protocol is described in the following:

- Number of members who attend the network is n.
- When member i has secret information x_i, each member calculates the following function keeping its secret information secret:
$$y = f(x_i, \cdots, x_n). \quad (1)$$
- All members can know y without disclosing their secret information at all.

The signature issuing methods using a multi-party protocol have already been proposed [3][4]. We construct a PKI on a pure P2P network using this multi-party protocol by members who compose a CUG (Fig.2). As issuing a certification corresponds to the digital signature technically, we can construct a CUG on a pure P2P network by issuing a certification using our multi-party protocol.

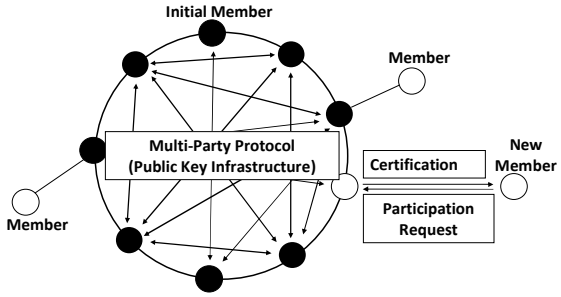

Fig. 2 Simplified P2P Network Certification Model

2.2 CUG construction method using multi-party protocol

Next, we explain a signature issuing method by a multi-party protocol. This protocol is basically composed of a protocol called Verifiable Secret Sharing (VSS) as illustrated in Fig.3. VSS is a method to divide secret information, and divided secret information can be verified whether they are genuine or not. We use a (k,n) threshold digital signature scheme that k out of n members must cooperate to decrypt a ciphertext [5]. We compose a random distribution method of secret information that creates information corresponding to a shared secret key and a public key in the network using the VSS scheme. Using this, we can construct the desired multi-party signature protocol that plural members sign to a specific data.

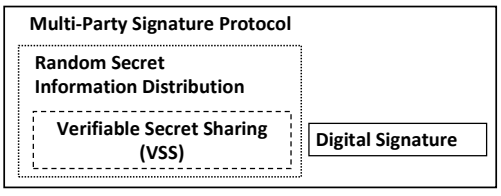

Fig. 3 Multi-Party Signature Protocol and VSS

Here, we define the following symbols:

- p : a very large prime number
- $a|b$: an integer b can be divided by an integer a
- q : a prime number which satisfies $q|p-1$
- Z_n : a set of integers that are more than zero and less than n
- Z_n^* : a set of integers which are included in Z_n and coprime to n
- e : a random number included in Z_q^* and is not zero
- g : a primitive root of p and is included in Z_p^*
- x : a secret key and is included in Z_q.

Now we explain a (k,n) threshold method as an example of VSS. Assume the dealer has a secret $s \in Z_q$ and is committed to s through public information $h = g^s \bmod p$. This secret can be distributed to $P_1,\cdots,P_n$ as follows:

PROTOCOL DISTRIBUTE (at the dealer)

step1: Choose a random polynomial

$$f(u) = f_0 + f_1 u + \cdots + f_{k-1}u^{k-1} \quad (2)$$

over Z_q of degree $k-1$ satisfying $f(0) = s$. Compute $s_i = f(i)$.

step2: Send s_i secretly to P_i and broadcast $(g^{f_i} \bmod p)_{i=1,\cdots,k-1}$ to all n participants.

Thus the dealer broadcasts $k-1$ elements in Z_p and sends secretly n elements in Z_q.

PROTOCOL VERIFY SHARE (at P_i)

step1: Verify that

$$g^{s_i} = \prod_{j=0}^{k-1} \left(g^{f_j}\right)^{i^j} \bmod p \,. \quad (3)$$

step2: If this is false, broadcast s_i and reject the dealer.

step3: For other each s_l claimed at step2, verify that

$$g^{s_l} = \prod_{j=0}^{k-1} \left(g^{f_j}\right)^{l^j} \bmod p \,. \quad (4)$$

If this is true, reject P_l. Otherwise, reject the dealer.

step4: If the dealer is not rejected, accept s_i.

Next we explain random secret information sharing scheme.

PROTOCOL RANDOM NUMBER (at P_i)

step1: Each P_i chooses $r_i \in Z_q$ at random and broadcasts $y_i = g^{r_i} \bmod p$ to all other participants.

step2: Each P_i distributes r_i by using PROTOCOL DISTRIBUTE. That is, P_i chooses a random polynomial such that

$$f_i(u) = r_i + a_{i,1}u + \cdots + a_{i,k-1}u^{k-1} \quad (5)$$

and sends $f_i(j) \bmod q$ to P_j secretly $(\forall j \neq i)$. P_i also broadcasts

$$g^{a_{i,1}}, \cdots, g^{a_{i,k-1}} \bmod p.$$

step3: Each P_i executes PROTOCOL VERIFY.

step4: Let $H := \{ P_j \mid P_j$ is not detected to be cheating at step 3$\}$. P_i computes

$$s_i := \sum_{j \in H} f_j(i) \text{ secretly.}$$

step5: Every P_i computes

$$y := \prod_{j \in H} y_j, \prod_{j \in H} g^{a_{j,1}}, \cdots, \prod_{j \in H} g^{a_{j,k-1}} \left(= g^{b_{k-1}}\right).$$

Proposition1:

In PROTOCOL RANDOM NUMBER, let

$$R := \sum_{j \in H} r_j, \quad f(u) := \sum_{j \in H} f_j(u).$$

Then,

$$y = g^R \quad (6)$$

$$f(u) = R + b_1 u + \cdots + b_{k-1}u^{k-1} \quad (7)$$

$$f(i) = s_i. \quad (8)$$

Key issuing protocol

step1: Each P_i executes PROTOCOL RANDOM NUMBER and gets

$$y(= g^x \bmod p), g^{b_1}, \cdots, g^{b_{k-1}} \bmod p$$

as a public output and also gets α_i as the secret output. And the public key is (p, q, g, y). Let $H_1 := \{P_j \mid P_j \in H$ and P_j is not detected to be cheating at step1$\}$.

Here,

$$\alpha_i = F_1(i), \quad (9)$$

where

$$F_1(u) = x + b_1 u + \cdots + b_{k-1}u^{k-1}. \quad (10)$$

If we assume that m is a signature target message, the signature issuing protocol will be constructed as follows.

Signature issuing protocol

Let m be a message and h be a one way hash function. Suppose that $B \subseteq H_l$ issue a signature.

step1: If $|B| < k$, stop. Otherwise, B execute PROTOCOL RANDOM NUMBER. Let the public output be

$$v(= g^e \bmod p), g^{C_1}, \cdots, g^{C_{k-1}} \bmod p$$

and the secret output of P_i be β_i. Let

$$w = v \bmod q. \quad (11)$$

From Proposition 1,

$$\beta_i = F_2(i), \quad (12)$$

where

$$F_2(u) = e + c_1 u + \cdots + c_{k-1}u^{k-1}. \quad (13)$$

Let $H_2 := \{ P_j \mid P_j \in B$ and P_j is not detected to be cheating at step1$\}$.

step2: If $|H_2| < k$, stop. Otherwise, each $P_i \in H_2$ reveals

$$\gamma_i := w\alpha_i + h(m)\beta_i \bmod q.$$

Here, γ_i is an element of certification and we can issue a certification using k pieces of γ_i.

step3: Each $P_i \in H_2$ verifies that

$$g^{\gamma_l} = \left(y \prod_{j=1}^{k-1} \left(g^{b_i}\right)^{l^j} \right)^w \left(v \prod_{j=1}^{k-1} \left(g^{c_j}\right)^{l^j} \right)^{h(m)} \text{ for } \forall l. \quad (14)$$

Let $H_3 := \{ P_j \mid P_j \in H_2$ and P_j is not detected to be cheating at step3$\}$.

step4: If $|H_3| < k$, stop. Otherwise, each $P_i \in H_3$ computes t satisfying

$$t = wx + h(m)e \bmod q \quad (15)$$

by applying the Lagrange interpolating formula to $\{\gamma_i\}$. The signature is (t, w) which corresponds to the signature target message m. (Remember that w is obtained at step1.) The validity of the signature (t, w) is verified by

$$w = \left(g^{t/h(m)} y^{-w/h(m)} \bmod p\right) \bmod q . \quad (16)$$

Here, γ_i is calculated from α_i and β_i which are proper to each member, and are not revealed to other members. So, γ_i can be calculated by only the owner of α_i and β_i, that is P_i. Consequently, t can be calculated by only members of the CUG.

Since we can now construct the signature using the multi-party protocol, we can also construct a CUG on a pure P2P network.

3 Reliability Issues of CUG construction method using multi-party protocol and proposal of certification issuing methods to new members

In this section, we study reliability issues of the CUG construction method using multi-party protocol and propose some methods of issuing a certification to a new member.

3.1 Reliability issues of CUG construction method using multi-party protocol

We consider a model that includes a group in which members have already been able to communicate securely with each other (Fig.2). In this model, some group can construct their own certificate authority for their group easily using a multi-party protocol. Members in this group communicate using certifications which are issued by the certificate authority. With this, it is possible to make space which can be accessed by only members of the group.

However, there are issues with reliability in this model in practical use. These issues are not negligible. First, members who are issued a certification cannot confirm whether the certification authority (CA) is reliable or not. Moreover, the certification authority (CA) is responsible for issuing a certification, however it cannot decide whether it should issue a certification or not by itself. In this paper, we resolve these issues by using the following assumptions. A group has its certification authority which is composed of its members. In addition, the certification authority issues a certification only when members in the group agree to issue it. The issued certification is only used for communication among the same group members. In this situation, confidence of the certification authority has an effect on only the group members.

Next, we study how to decide to issue a certification to a new member by members who have already been in the CUG. The basic policy is that each member decides whether the certification authority should issue a certification or not independently and the certification authority in the group decides whether it issues a certification or not based on the result. Concretely, each member indicates whether it agrees to issue a certification or not. Using the decisions from all members, we consider how to decide to issue a certification based on many policies such as decision by a majority or a portion of approval in all votes or unanimous vote. In this paper, we consider the method that fulfills the following two requirements.

Requirement 1 If the number of members who decide whether it approves to issue a certification to a new member or not is n and number of members who approve to issue a certification is $l(\leq n)$, then the certification will be issued.

Requirement 2 Each member has no knowledge of the other members' vote.

These two requirements correspond to flexibility and anonymity of a decision method respectively. The requirement 1 is necessary to provide a flexible decision method to each group in the view point of service. The requirement 2 is necessary because it is preferable to avoid that each member's decision is revealed to other members in order to manage the group smoothly.

In the following three subsections, we propose achievable decision policies for issuing a certification and evaluate whether they meet above-mentioned requirements or not and study what is necessary to achieve them.

3.2 Alternative multi-party signature protocol

We can consider the method that puts the alternative model that each member chooses either approval or disapproval and each member's vote is hidden from other members into a multi-party signature protocol

such as Fig.4. If this method is realized, it will be possible to achieve the goal that satisfies two requirements since the protocol is proceeded without showing whether information provided by members is adequate for issuing a signature or not and the result that a true signature will not be issued shows that information provided by members is not adequate. As this method cannot decide whether data for issuing a signature is true or not, when collected divided information is more than necessary for issuing a signature, the signature may be issued or not issued according to the choice from collected divided information. Consequently, a divided number of information should be the same as necessary number for issuing a signature. In this method, if not any plural members but only one member disapproves issuing a signature, the correct signature will not be issued. In other words, this method is constructed unanimously. If this method is implemented, we can provide a protocol which meets the requirement 1 as unanimous construction and meets the requirement 2 fully. As this method can achieve both decision of a member for issuing a signature and procedure for actual issuing a signature only by one protocol, it is easy to discuss its safety and we consider this method as the applicable and useful method.

However, this method requires the strict condition that a divided number of information should be the same as necessary number for issuing a signature. So, in the following two subsections, we propose the methods that do not require this strict condition.

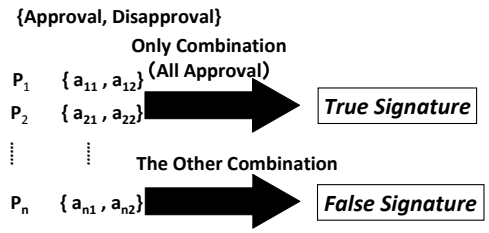

Fig.4 Alternative Multi-Party Signature Protocol

3.3 Deviation from multi-party signature protocol

We think that the easiest decision policy that does not require the above-mentioned strict condition is to let a member who disapproves to issue a certification deviates from a multi-party signature protocol intentionally or does not transmit necessary information. This method can meet the requirement 1 but cannot meet the requirement 2. In a multi-party signature protocol, when information of k members is derived, issuing a signature will be possible. Here, k can be set at will when the protocol is constructed. As this means that l can be set at will, it is clear that this method meets the requirement 1. However, it is possible for this protocol to check whether information that has been shown by all members is right or not to make a signature and both members who achieved the protocol correctly and members who deviated from the protocol are revealed, so this method does not meet the requirement 2.

3.4 Collection point model

If a variable which corresponds to an identifier of some member cannot be related to a specific member in subsection 3.3, even when the method deviates from a multi-party signature protocol, it seems to be possible for this method to meet the requirement 2.

We assume that an identifier (ID) of a member is an integer. This ID is used in the following situations concretely. First, in the first step of information division of random secret distribution, IDs whose owners are not information creator will be needed to calculate information division creation (i.e. calculation of f (ID)). Next, when the member presents divided information or its own open information by itself, it must send the information with its own ID in order to show that it created the information. In the multi-party signature protocol, in order to create the right signature, the ID of some user must be always identical. In other words, each member needs to know the receiver's ID for checking or creating a signature and it must inform its ID to the receiver. For this reason, the ID of the member who deviated from the protocol will be known to every member in the process of verification.

Though the ID is a type of identifier, it is necessary to know its own information such as a corresponding IP address on the Internet. Actually, there are methods that relate the ID with the IP address. Here we study the method that enables data exchange without relating the ID with the IP address contrarily. In other words, we suppose the situation that some member knows the ID of the data receiver, but he does not know its actual IP address. Namely, we suppose that the data transmitter sends its packet

with its own ID but the receiver of the data does not know the actual IP address of the transmitter. In this scenario, even when a member deviates from the protocol, the other members know only its ID.

Here, not relating the ID with the IP address does not mean to veil approval or disapproval of a member to issue a signature completely. This is because the identity of a member will be revealed by continuing to use the same ID even when its IP address is not revealed to the public. This situation will be solved by reconstructing IDs of members as random IDs frequently. From the above, if the ID is not related to its own information such as an IP address and the ID can be changed often, it is supposed that veiling approval or disapproval of each member is achieved even when deviating from the protocol is regarded as disapproval for issuing a signature.

So, we suppose that a collection point as shown in Fig.5 is set on the network. First, we prepare the list which relates public keys with IDs in advance. When some member wants to transmit some data to an ID, it encrypts the data with the receiver's public key and transmits it to the collection point. Each member retrieves information whose destination is it and public information and decrypts the information for it using its own secret key. By this, it will be realized to exchange data without knowing the other member's IP address. Of course, the collection point must not be any specific terminal but constructed in a distributed manner on the network. As an example of managing data in a distributed manner on a P2P network, a Distributed Hash Table (DHT) is known and it is an achievable technology. We suppose that we can make it difficult for group members to know the transmitter or the receiver of the data by constructing a collection point from not only group members but also all members in the pure P2P network.

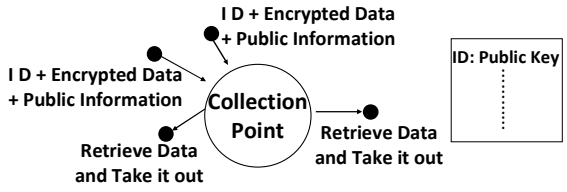

Fig.5 Collection Point Model

However, this method has the following issues. The biggest issue is to make a list which relates public keys with IDs. This list should be made in the same structure as a randomly secret distribution method. In other words, all members can share a pair of public key and ID commonly without making each member's ID public. If this is achieved, this method can make use of the above-mentioned way and meets both the requirement 1 and the requirement 2. In addition, as an essential issue of this method, it may put a heavy load on the entire network more than necessarily because it stores data in other place once. For this reason, the method which efficiently distributes and collects data which are used for issuing a signature is needed in order to realize the collection point model.

4 Strong points and applications of collection point model

In this section, we discuss strong points and applications of the collection point model which we discussed in 3.4.

4.1 Strong points of collection point model

The first strong point of the collection point model is flexibility in communication. This flexibility means that both members who communicate in this model need not set the time to communicate. In other words, this model does not require that both members who communicate must exist online simultaneously. This flexibility has the following good points for both a data transmitter and a data receiver. For a data transmitter, when it wants to transmit some data to some receiver, it can encrypt the data with the receiver's public key and transmit it to the collection point whether the data receiver is online or not. On the contrary, for a data receiver, it can retrieve and take out data whose destination is it and public information and decrypt them using its own secret key even when the data transmitter is not online.

The second strong point of the collection point model is pseudonym communication. This means that members in the collection point model can transmit data without knowing a receiver's real IP address and on the contrary a receiver can retrieve and take out data without knowing a transmitter's IP address.

We can regard this collection point model as one of a means of realizing I2P (Invisible Internet Project) anonymous network [6]. This I2P is an anonymous network, exposing a simple layer that

applications can use to anonymously and securely send messages to each other. The I2P project was formed in 2003 to support the efforts of those trying to build a more free society by offering them an uncensorable, anonymous, and secure communication system. I2P is a development effort producing a low latency, fully distributed, autonomous, scalable, anonymous, resilient, and secure network. The goal of I2P is to operate successfully in hostile environments - even when an organization with substantial financial or political resources attacks it. In I2P, the first time a client wants to contact another client, they make a query against the fully distributed "network database" - a custom structured distributed hash table (DHT) based off the Kademlia algorithm [7]. Namely, the network database in I2P works as the same as the collection point in the collection point model.

By this second strong point, members in the collection point model will not suffer Denial of Service (DoS) attacks directly. Moreover, privacy of members in the collection point model is protected.

Moreover, the collection point model has also inherited P2P network's strong points. The first strong point inherited from P2P network's strong points occurs from the nature that data are stored not in centrally but in distributed manner. In a client/server environment that stores data in the server centrally, when a client needs data, it must retrieve them from the server at that time. On the other hand, in P2P network, as data are stored in distributed manner, each member can retrieve and take out data in background beforehand and prepare to use them. In P2P network, of course also in the collection point model, a fixed server like a POP server does not exist but each member performs its function instead of it.

Next, the second strong point inherited from P2P network's strong points is a high scalability. As the collection point that stores data is not located centrally in a specific member node but is located distributed in the P2P network, even when stored data in the collection point become huge, its effect to each member node is not too big.

4.2 Applications of collection point model

The collection point model is applicable to many kinds of ballot systems. For example, this model is applicable to a vote for confidence of a director in some society or organization, because this type of vote is casted by members of the society or organization and members only have to cast either a vote in favor of confidence or a vote against confidence. Besides, this type of vote does not require real-time characteristics and usually a period of voting is set. So each member can cast its vote whenever it accesses the P2P network during a period of voting.

Another example is a vote casted by members in many regions all over the world. In this example, members cannot cast a vote simultaneously because of time lag. However, in the collection point model each member can cast a vote whenever he or she likes.

Moreover, the collection point model is applicable to not only many kinds of ballot systems but also pseudonym communication system like I2P that we introduced in 4.1.

In other words, the collection point model can realize the following internet applications which are provided by I2P:

- Web browsing: using any existing browser that supports using a proxy.
- Chat: IRC, Jabber, I2P-Messenger.
- File sharing: I2PSnark, Robert, iMule, I2Phex, PyBit, I2P-bt and others.
- E-mail: susimail and I2P-Bote.
- Blog: using e.g. the pebble plugin or the distributed blogging software Syndie.
- Distributed Data Store: Save your data redundantly in the Tahoe-LAFS cloud over I2P.
- Newsgroups: using any newsgroup reader that supports using a proxy.

Therefore, the collection point model has many applicable fields.

5 Conclusion

In this paper, we studied a method of constructing a Closed Users Group (CUG) on a pure P2P network that has both high fault-tolerance and scalability using a multi-party protocol without any administration methods. First, we showed that a Public Key Infrastructure (PKI) can be constructed on a pure P2P network using the multi-party signature protocol. In addition, we showed that it is possible to implement a CUG using a PKI. Finally, we studied the decision method of issuing a certification to a new member.

In selecting the decision method to issue a signature, we look into a number of methods in order to achieve flexibility and keep anonymity at the same time. As a result, we proposed the alternative multi-party signature protocol and the method that veils sender/receiver (the collection point model).

In addition, we discussed strong points and applications of the collection point model. We

concluded that strong points of the collection point model are flexibility in communication, pseudonym communication, short data retrieval, and high scalability. The latter two strong points are the P2P networks' strong points. And we concluded that applications of the collection point model are not only many kinds of ballot systems but also pseudonym communication system like I2P. Therefore the collection point has many applicable fields.

References:

[1] T. Shizuno, T. Kitamura, and T. Okabe, Decentralized Authentication Method for P2P Network, *IEICE B-6-29*, pp.29, March 2007.

[2] Blum, M., Coin Flipping by Telephone, *IEEE, COMPCON*, pp.133-137, 1982.

[3] Manuel Cerecedo, Tsutomu Matsumoto, Efficient and Secure Multiparty Generation of Digital Signatures Based on Discrete Logarithms, *IEICE Trans. on Fundamentals* , vol.E76-A, no.4, pp.532-545, 1993.

[4] Choonsik PARK, and Kaoru Kurosawa, New ElGamal Type Threshold Digital Signature Scheme, *IEICE Trans. on Fundamentals* , vol. E79-A, no.1, pp.86-93, Jan. 1996.

[5] T.P. Pedersen, Distributed Provers with Applications to Undeniable Signatures, *Proc. Eurocrypt'91 Lecture Notes in Computer Sciences*, LNCS 547, pp.221-238, Springer-Verlag, 1991.

[6] I2P Anonymous Network, available at http://www.i2p2.de/index.html.

[7] Kademlia: A Peer-to-peer Information System Based on the XOR Metric, available at http://pdos.csail.mit.edu/~petar/papers /maymounkov-kademlia-lncs.pdf

Masao Tanabe

Received his B.E. and M.E. degrees in Electronics and Communication Engineering from Waseda University, Tokyo, Japan, in 1985 and 1987, respectively. He joined NTT Laboratories in April 1987. He is also a graduate student of the Graduate School of System Design, Tokyo Metropolitan University. His current interests include security issues in communication networks. He is a member of the IEEE and the IEICE.

Keita Sugiyama

Received his B.E. degree in Information Systems Engineering from Tokyo Institute of Technology, Japan, in 2006. He received his M.E. degree in System Design from Tokyo Metropolitan University, Japan, in 2008. He joined Nomura Research Institute, Ltd. in 2008. He is a member of the IEICE.

Masaki Aida

Received his B.S. and M.S. in Theoretical Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, respectively, and received the Ph.D. in Telecommunications Engineering from the University of Tokyo, Japan, in 1999. In April 1989, he joined NTT Laboratories. From April 2005 to March 2007, he was an Associate Professor at the Faculty of System Design, Tokyo Metropolitan University. He has been a Professor of the Graduate School of System Design, Tokyo Metropolitan University since April 2007. His current interests include traffic issues in computer communication systems. He received the Young Researchers' Award of IEICE in 1996. He is a member of the IEEE, the IEICE, and the Operations Research Society of Japan.