Realizing and Evaluating Mutual Anonymity in P2P Networks

Chigusa Kawashima, I. G. B. Baskara Nugraha, Hiroyoshi Morita
Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
Email: {chigusa@appnet., baskara@, morita@}is.uec.ac.jp

Todorka Alexandrova
Department of Computer Science
Waseda University
Tokyo, Japan
Email: alexandrova@aoni.waseda.jp

***Abstract*—In this paper we propose a mutually anonymous protocol for decentralized Peer-to-Peer (P2P) networks. The protocol is a combination between the Secret-Sharing-Based Mutual Anonymity Protocol (SSMP) and the information slicing technique. The proposed protocol realizes the initiator's and responder's anonymity by using the SSMP in which the complete reply-confirm interaction between responders and initiators is realized using the information slicing algorithm. Employing the concept of secret sharing schemes plays an essential role for the protection of the transmitted information between the initiator and responder, and using the information slicing technique the proposed protocol is churn resilient and can be realized with lower cryptographic cost. Moreover, we evaluate the anonymity in the P2P system from probability point of view. The results show that the proposed mutual anonymity protocol provides higher anonymity than the conventional methods.**

I. Introduction

Peer-to-Peer (P2P) networks have become quite popular in the recent years for their easy resource sharing and powerful search scheme. Thus, with their development the anonymous communication has become an important issue.

Han et al. [1], [2], [3], presented a *Secret-sharing-based Mutual Anonymity Protocol (SSMP)*, which guarantees mutual anonymity in decentralized P2P networks. The mutual anonymity assures the initiator's anonymity, the responder's anonymity and the anonymous communication between them, i. e., the initiator and the responder do not know each other. The SSMP uses *a secret sharing scheme* [6] to guarantee anonymous query issues and the *onion routing technique* [7] to achieve a complete reply-confirm interaction between responders and initiators. Each potential responder, that is able to provide the requested file, creates two onion routing paths, one for sending the reply information from the responder to the initiator and the other for the initiator to deliver the confirmation message to the responder.

However, the onion routing method has some drawbacks. In the onion routing algorithm, the initiator has to collect all the IP addresses and public keys of all the nodes in the overlay. Then the message is encrypted in layers with the public keys of the nodes along the path. The initiator's and responder's anonymity is realized by sending the route setup message, wrapped in layers of public key encryption, through a chain of nodes. Hence, a trusted public key infrastructure (PKI) is required to distribute the public keys of each node in the P2P network, which makes the system vulnerable to security breaches and less efficient. However, in the onion routing mechanism the initiator is aware of the responder's IP address, while the responder does not know who the initiator is, i.e., it doesn't provide mutual anonymity in the system.

To solve the PKI requirement problem, Katti et al. [4], [5], proposed the *Information Slicing* technique, which realizes not only anonymous communication, but also acquires churn resilience without using any PKI. Thus, the information slicing method is an alternative way to realize the anonymity addressed by the onion routing idea without using public key cryptography. Instead of creating an anonymous path as in the onion routing idea, the information slicing method sends each intermediate node its routing information (i. e., its next hop's IP address) in a confidential message sliced over multiple disjoint paths. However, the information slicing algorithm doesn't guarantee mutual anonymity as well.

In this paper we propose mutually anonymous communication protocol, which is a combination between the SSMP and the information slicing technique. The proposed protocol realizes the initiator's and responder's anonymity by using the SSMP in which the complete *reply-confirm* interaction between responders and initiators is realized using the information slicing algorithm and thus, PKI is not required. In the proposed scheme the responder and the initiator are able to interact with each other even though they are not aware of each others IP addresses. Employing the concept of secret sharing schemes plays an essential role for the protection of the transmitted information between the initiator and responder, and using the information slicing technique the proposed protocol is churn resilient and can be realized with lower cryptographic cost. An evaluation of the anonymity in the P2P system from probability point of view has been performed. The results show that the proposed mutual anonymity protocol provides higher anonymity than the conventional methods.

The rest of this paper is organized as follows. In the next section we present the basic idea behind threshold secret sharing schemes. Section 3 and Section 4 describe the scenario of the onion routing method and the information slicing algorithm. The proposed protocol is described in Section 4 and in Section 5 we make evaluation of the anonymity. The last section summarizes the results in this paper and suggests directions for future work.

II. Secret Sharing Schemes

Threshold secret sharing schemes were introduced by Shamir in 1979 [6]. A secret sharing scheme is called a (t, n) threshold secret sharing scheme if knowledge of any t or more shares makes the secret s computable and the knowledge of

978–1–4244–6014–4/10/$26.00 ©

any $t-1$ or fewer shares leaves s completely undetermined (in the sense that all its possible values are equally likely).

Let $D \notin P$ be a dealer who selects a secret $s \in \mathcal{K}$ and distributes shares $s_i \in \mathcal{S}$ to each user $P_i \in P$, where $P \triangleq \{P_1, .., P_n\}$.

Shamir's (t, n) threshold secret sharing scheme is described as follows [6]:

1. Let $\mathcal{K} = GF(q)$ and $\mathcal{S} = GF(q)$, where q is a prime power and $q \geq n+1$.

2. D chooses elements $a_1, \ldots, a_{t-1} \in GF(q)$ independently and uniformly, and constructs the polynomial $f(x)$:

$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}. \quad (1)$$

3. D chooses n distinct points $x_i \in GF(q)$, $1 \leq i \leq n$. The values x_i are public.

4. D distributes shares $s_i = f(x_i)$ to users P_i, $1 \leq i \leq n$.

Using polynomial interpolation every t or more users can recover the secret s and any group of $t-1$ or less users obtains no information about the secret.

III. Onion Routing

The onion routing method is a routing technique which realizes initiator's anonymity using public key cryptography [7]. In this method, we assume that the initiator knows the IP addresses of some nodes in the network which are utilized as intermediate nodes and also it knows the public keys of the intermediate nodes.

At first, the initiator picks randomly relay nodes to the responder from the network. For example, the initiator chooses nodes A, B, C, as described in Figure 1.

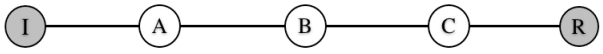

Fig. 1. Path of Onion Routing Method

The initiator encrypts the message to be delivered using each intermediate node's public key. We denote node i's public key as K_i, and the encryption of the message m using K_i as $E_{K_i}(m)$. Before message transmitting, the initiator encrypts the message m as follows:

$$E_{K_A}\{B'\text{s ad}, E_{K_B}\{C'\text{s ad}, E_{K_C}\{R'\text{s ad}, E_{K_R}(m)\}\}\}.$$

The next node's IP address is encrypted with the previous node's public key.

In the decryption phase, the initiator sends the encrypted message to node A. Node A receives the message and then decrypts it with its own secret key, and obtains node B's IP address and the remaining encrypted message which is encrypted with B's public key. Then, A forwards it to B and B also decrypts the message and obtains C's IP address. In this way, each intermediate node obtains its next node's IP address and finally the responder receives the message m. Each intermediate node only knows its previous and next nodes' IP addresses. Note that first intermediate node knows the initiator's IP address. However, the first intermediate node can not distinguish whether its previous node is the initiator or one of the intermediate nodes. Similarly, each intermediate node can not distinguish the destination of the message.

A drawback of the onion routing method is the need for PKI. PKI is problematic for P2P anonymizing networks because key distribution and management are well known difficult problems since it opens up the system to attacks in the key distribution procedure. Moreover, in a large P2P network, the trust model may differ from one node to another. Employing PKI makes the system inefficient and not scalable.

IV. Information Slicing Algorithm

In [4], [5], Katti et al., proposed an algorithm, which realizes the onion routing idea without using any public key cryptography. This algorithm is based on the idea of information slicing. A detailed description of the information slicing routing mechanism can be found in [4].

The main idea behind the information slicing method is first to randomize the message and then divide the randomized message into pieces that are sent along disjoint paths to the destination. Only the destination receives all the pieces and can decode the message. Instead of creating an anonymous path, the information slicing method sends each intermediate node its routing information, such as its next hop's IP address, in a confidential message sliced into multiple pieces and delivered over multiple disjoint paths.

Information Slices Construction. Let Alice want to send the message $\mathbf{m}$ to Bob. First, Alice divides the message into d pieces, i.e., $\mathbf{m} = (m_1, \ldots, m_d)$ and multiplies it with a random, invertible $d \times d$ matrix $\mathbf{A}$:

$$\mathbf{I} = \begin{pmatrix} I_1 \\ \vdots \\ I_d \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_d \end{pmatrix} \mathbf{m}^T = \mathbf{A}\mathbf{m}^T, \quad (2)$$

where $\mathbf{m}^T$ stands for the transposed column vector of $\mathbf{m}$.

Then, Alice picks d disjoint overlay paths to Bob and sends on path i the slice I_i and $\mathbf{A}_i$, which stands for the i'th row of the matrix $\mathbf{A}$. Upon receiving all the slices $I_1, \ldots, I_d$, Bob can decode the original message as follows:

$$\mathbf{m}^T = \mathbf{A}^{-1}\mathbf{I}, \quad (3)$$

where $\mathbf{A}^{-1}$ stands for the inverse matrix of matrix $\mathbf{A}$.

Constructing the Forwarding Graph. An example how the forwarding graph is constructed is shown in Figure 2. We assume the initiator has access to two IP addresses I and I', that have a secure connection between them. The initiator picks randomly the relay nodes (V, W, X, Y, Z) from a set of available nodes and arranges them into $l = 3$ stages (excluding I and I') each containing $d = 2$ nodes. The receiver R is randomly assigned to one of the stages in the graph.

The initiator has to send each relay node the IP addresses of its next hops by splitting them into two slices. To do so he divides the IP addresses into two parts $\mathbf{V}_l$ and $\mathbf{V}_h$, that stand for the low and high words of the address of node V, respectively. Then the two slices are obtained as follows

$$\begin{pmatrix} \mathbf{V}_L \\ \mathbf{V}_H \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{V}_l \\ \mathbf{V}_h \end{pmatrix}. \quad (4)$$

Figure 2 shows how the slices are forwarded so that each node knows only the IP addresses of its children. For example,

once V receives its slices, it can recover the IP addresses of X and Y:

$$\begin{pmatrix} \mathbf{X}_l & \mathbf{Y}_l \\ \mathbf{X}_h & \mathbf{Y}_h \end{pmatrix} = \mathbf{A}^{-1} \begin{pmatrix} \mathbf{X}_L & \mathbf{Y}_L \\ \mathbf{X}_H & \mathbf{Y}_H \end{pmatrix}. \quad (5)$$

However V can not recover the IP addresses of X's and Y's children form the slices it receives. Using the available relay nodes the initiator can construct the forwarding graph so that each information slice is routed to the respective nodes along vertex disjoint paths and a detailed algorithm for this is given by Katti et al. [4].

In addition to sending each node its next hop IPs, the initiator sends a symmetric key and a flag, indicating the receiver. The key and the flag are also split into slices and can not be recovered from any intermediate node. Moreover, the receiver shares a secret key with the initiator. The initiator encrypts the message with this key, splits it into slices and forwards it along the constructed graph. All the relay nodes can see the encrypted message but only the receiver can decrypt it.

More details on the information forwarded to each node in the graph establishment step and the data transmission step can be found in [4] and are omitted here.

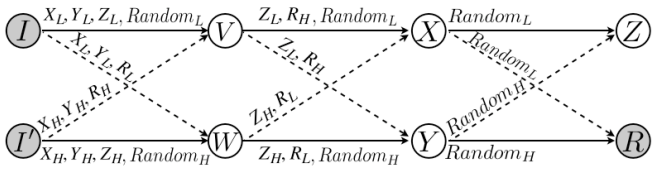

Fig. 2. Information Slicing

V. Mutually Anonymous Protocol

In this section we describe the proposed mutually anonymous communication protocol, which is a combination between the SSMP and the information slicing technique. The protocol consists of three main steps: anonymous query step, reply and confirm message step and file delivery step, that are described here.

Basic Notations:

- I - the initiator;
- R - the responder;
- F - a file to be transmitted;
- f - the ID or the name of a file F;
- sq - the sequence number to mark a query;
- is_R - the information sequence of R;
- K_P - the public key of a peer P;
- $E_{K_P}(\cdot)$ - encryption with P's public key.

Note that in the proposed protocol, each public key does not need to be authenticated by PKI because the initiator does not collect the public keys of other users preliminarily.

A. Step 1: Anonymous Query

In this step the initiator I performs an anonymous query for a file F with ID f. This is done following the algorithm described in [3] and can be described in two stages as follows.

(i). First the initiator divides the file ID, f, using a (t, n) threshold secret sharing scheme, into n shares: $f_1, f_2, \ldots, f_n$. Then he creates a sequence number sq to mark the query and randomly selects a list of neighboring peers $N_1, N_2 \ldots N_n$. To each of these neighboring peers N_i, $1 \leq i \leq n$, he sends the packets consisting

$$s_i = \{f_i, sq, t, K_I\}, \quad 1 \leq i \leq n. \quad (6)$$

When each I's neighbor receives a new query share, it broadcasts the share in probability p or forwards it to a randomly chosen neighbor in probability $(1 - p)$.

The described stage is illustrated in Figure 3.

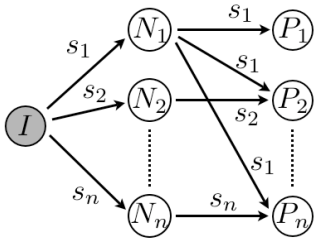

Fig. 3. **(i)**. Anonymous Query

(ii). For both broadcasting and forwarding, each peer keeps an entry of each share in its *sequence number routing table* as described in Table 1. The sequence number routing table consists of the IP addresses of the nodes from which (source nodes) a query with sequence number sq has been received. For each query the peer marks the distinct shares with the sequence number sq. When a peer collects at least t shares with the same sequence number, then it can recover the original query. This is done using the secret recovering algorithm for the (t, n) threshold secret sharing schemes given in Section 2.

Sequence Number	Source IP Address
sq1	IP1,IP2,...
sq2	IP5,IP6,...
...	...

Table 1. Sequence Number Routing

Without loss of generality we assume that P_1 is one of the peers that collects t shares $s_{i_1}, s_{i_2}, \ldots, s_{i_t}$ and recovers f. When P_1 obtains the recovered f then it floods the original query for I and sends the packets consisting

$$\{f, sq, P_1\text{'s IP}, K_I\}. \quad (7)$$

Thus, P_1 becomes I's *agent* and in this way the original query reaches a peer R that is able to provide the requested file f and also P_1 is not aware of $I's$ IP address. There might be more than one agents for a given file query in the system, but without loss of generality, in order to make the explanation clear we assume it is just one.

In this step an anonymous query for a file with ID f has been performed and it is illustrated in Figure 4.

B. Step 2: Reply and Confirm Messages

In this step the responder R and the initiator I anonymously exchange their *reply* and *confirm* messages for the requested file f. In [3] this is done by constructing two onion routing paths. In this paper we propose a way to realize this reply and confirm interaction between the initiator and responder by replacing the onion routing paths by *information slicing routing paths*, proposed by Katti et al., [4].

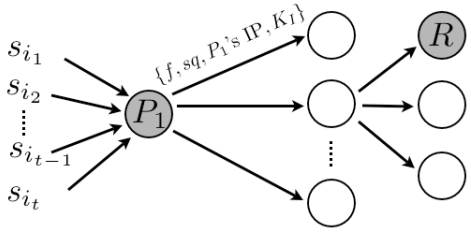

Fig. 4. **(ii)**. Query Flooding

1) Reply Message: The responder R anonymously delivers the reply message to I through P_1. For that purpose R constructs information slicing routing path to P_1. This is possible, since R has already received P_1's IP address as described in (7).

There might be more then one responders R that are willing to share the requested files, such as R_1, $R_2 \ldots$, and so on. In order to distinguish the information slicing routing paths to P_1 coming from the different potential responders, each responder R, R_1, $R_2, \ldots$, generates randomly its own identification sequence information sequences is_R, is_{R_1}, $is_{R_2}, \ldots$, corresponding to the query sq.

This sequence is just used to identify the information slicing path and knowing it the peers are not able to recover $R's$ IP address. It can be thought as the sequence number identifying the information slicing path from R to P_1.

Together with the graph establishment information that R sends to each node $(U, U', V, V', \ldots, P_1, P_1')$ as described in [4], it also sends its information sequence is_R to each of the nodes along the path. Then each node in the path makes its *slicing routing table*, which keeps the IP addresses of the nodes from which it received slices with information sequences is_R, is_{R_1}, $is_{R_2}, \ldots$ for a query sq. The idea of the information slicing table is illustrated in Table 2.

is	Source IP Address	Destination IP Address
is_R	IP1,IP2,...	IP3,IP4,...
is_{R_1}	IP5,IP6,...	IP7,IP8,...
...	...	...

Table 2. Slicing Routing

The data that R is willing to send anonymously to P_1 is:

$$\{reply, sq\}. \tag{8}$$

The *reply* message consists of description of provided files, the identity sequence of R, is_R and the public key K_R. When the reply packet reaches P_1, it firsts decrypts it with its symmetric key and then obtains the file name f and the sequence number sq. Then it checks the sq in its routing table. P_1 integrates all the received reply messages together with their identifying sequences into a single reply message $Reply\_P_1$, encrypts it with $I's$ public key K_I and marks it with sq and another sequence number sq_P_1, which distinguishes reply shares from other agent peers. Then P_1 delivers this reply to I along the reverse path based on sq. Each intermediate peer that receives the packet checks sq in its routing table and delivers the packet to the peer that sent the corresponding query message to it. If there is no entry marked sq the peer discards the packet. Following this mechanism the reply message reaches I.

The described step is illustrated in Figure 5.

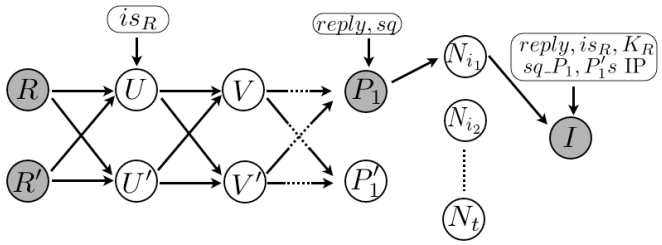

Fig. 5. Reply Message

2) Confirm Message: After receiving the reply message from P_1, the initiator picks a peer as a responder. Without loss of generality we assume it choses R with identifying sequence is_R. Then it constructs an information slicing routing path to P_1 with nodes $(I, I', X, X', Y, Y', \ldots)$. Together with the slices that build the graph, I sends to each node the information sequence is_R and to node P_1 it sends the data that consists of the confirm message encrypted with R's public key, i.e.,

$$\{E_{K_R}(confirm)\}. \tag{9}$$

Analogically to the graph constructed for the reply message, in this case each node also makes its slicing routing table for is_R. Thus, the information sequence is_R becomes the identification of both of the paths constructed between R and P_1, and P_1, and I.

After P_1 obtains is_R it checks is_R in its local slicing table and forwards the encrypted message $\{E_{K_R}(confirm)\}$ to the peers corresponding to is_R. The peers in the previous hop do the same and eventually using the reverse information slicing route the encrypted confirm message reaches R. Using its private key R is able to decrypt it.

The described step is illustrated in Figure 6.

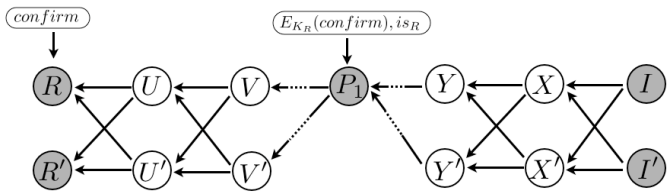

Fig. 6. Confirm Message

C. Step 3: File Delivery

Upon the received confirm message R delivers the desired file to I through the two slicing paths already constructed between R and P_1, and P_1, and I. The agent P_1 plays the connection role between them.

R encrypts the file F with I's public key K_I and shares $E_{K_I}(F)$ using a $(2, 2)$ threshold secret sharing scheme for the example given in Figures 5 and 6. The obtained shares are F_1 and F_2. Then it delivers the shares to P_1 using the path with the nodes

$(U, U', V, V', \ldots, P_1', P_1)$. P_1 delivers the shares F_1 and F_2 to I along the reverse path $(Y, Y', X, X', \ldots, I', I)$ using the information of the slicing routing tables for is_R of each node along the path. Even though all the nodes obtain the both of the shares F_1 and F_2 they are not able to recover the file F. The only one that is able to do so is the initiator I, using its private key.

The file delivery step is illustrated in Figure 7.

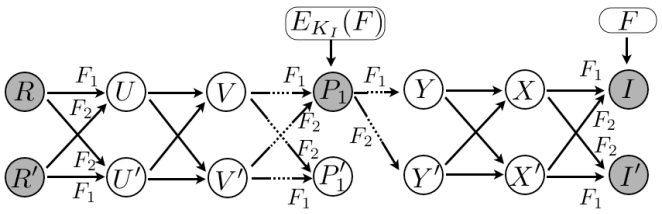

Fig. 7. File Delivery

It is worth noting that in the proposed protocol the existence of an agent P_1 plays an essential role in the mutual anonymity realization and employing the concept of secret sharing schemes plays an essential role for the protection of the transmitted information between the initiator and responder. Moreover, using the information slicing technique the proposed protocol is churn resilient and can be realized with lower cryptographic cost.

VI. Evaluation of Anonymity

We evaluate the anonymity of the onion routing method and the information slicing method, and then we apply it in order to make the comparison between the proposed protocol and SSMP. In our protocol we replace the onion routing method used in the SSMP with the information slicing technique. That is why the comparison of the anonymity of the onion routing method and the information slicing method is essential for the evaluation of the proposed protocol.

Here we discuss the probability that a set of malicious nodes can identify the source/destination in the anonymizing network by observing fractions of network traffic and colluding among themselves. We do not consider global attacker who can snoop on all links in the network, and we also assume that the attacker can not snoop on all paths leading to destination.

Let p_m be the fraction of malicious nodes in the anonymizing network ($p_m = \frac{c}{n}$, where n is the number of nodes in the network and c is the total number of malicious nodes). In case that the malicious nodes can identify the source/destination completely, we say that source/destination's anonymity fails.

A. Source/Destination Failure in Information Slicing Method

Katti et al. give evaluation of source anonymity failure and destination anonymity failure on the basic information slicing method in [4]. We evaluate the proposed protocol using a similar method.

If all the nodes in stage 1 (note that stage 0 is the source, i.e., the nodes in stage 1 are neighbors of the source) are malicious, the anonymity of the source is 0. The reason is that the attacker can decode the entire graph, discover that it controls the first stage, and thus the previous stage has to be the source stage. Thus, the probability of source failure is:

$$P(SrcFailure) = p_m^d. \tag{10}$$

Suppose that all the nodes in some stage i upstream of the destination are malicious. Then the malicious node can decode the downstream graph and discover the intended destination. Assume the destination is in stage $j+1$. Then the probability that at least one entire stage before the stage $j+1$ consists of malicious nodes is given by

$$P_{fail}(j+1) = 1 - \{1 - p_m^d\}^j. \tag{11}$$

Since the destination could be in any stage with equal probability $1/l$, the overall probability is:

$$P(DstFailure) = \frac{1}{l}\sum_{j=1}^{l-1} P_{fail}(j+1). \tag{12}$$

B. Source Failure in Onion Routing Method

Syverson [9] defines the probability that the malicious nodes can identify source and destination on the onion routing path. In this scenario, the intermediate nodes on the onion routing path see various consistent latencies between nodes. This might be possible if packet decryption and encryption dominated the message latency. Per-hop delay might be very consistent in some LANs. Timing analysis in this scenario would reveal to the two attacker nodes that they were on the same path and reveal the number of hops between them. Given that the path length is known, the attacker will know if the first attacker node follows the initiator directly. In this way, if the attackers compromise both the first and last node on the path, they will immediately identify the initiator. Thus, this attack succeeds with probability p_m^2.

C. Evaluation Using Chernoff bound

We compare the anonymity of the information slicing method with the onion routing method using Chernoff's inequality. M. Wright et al. [8], give analysis of the predecessor attack in anonymous communications system. They evaluate the onion routing method depending on how many times of path reforming are needed to identify the initiator. In this section, we evaluate the anonymity of the information slicing method in a similar way as M. Wright et al., and compare its anonymity to the anonymity of the onion routing method.

Let T be the number of times that an initiator is reforming its anonymous path. For example, if an initiator reforms the anonymous path to a responder five times (note that the responder is not changed and intermediate nodes are not fixed), in this case $T = 5$. In addition, we call path reforming *round*.

We calculate the bound of T by the probability of the attack success in a single round. We have already obtained the single round probability as described above, i.e., the single round probability of the onion routing method is p_m^2 as given in Section VI-B, and the single round probability of the information slicing is p_m^d as given in Section VI-A.

1) Chernoff bound: To obtain the bound of T, we employ the Chernoff bound [10]. Let X be a random variable. Then, the Chernoff bound is described as follows:

$$P[X \leq A] \leq \inf_{t<0} e^{-tA}\mathrm{E}[e^{tX}], \tag{13}$$

where $\mathrm{E}[e^{tX}]$ is the expected value of e^{tX}. In the case of X having a binomial distribution, the following bound is satisfied

for $A = (1-\delta)\mu$

$$P[X < (1-\delta)\mu] < \left[\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right]^{\mu}, \quad (14)$$

where $\mu = \mathrm{E}[X]$ is the expected value of X, and δ is an arbitrary number ($0 < \delta \leq 1$). We can simplify this by noting that $(1-\delta)^{(1-\delta)} > \exp(-\delta + \delta^2/2)$, and then we have

$$P[X < (1-\delta)\mu] < e^{-\frac{\delta^2}{2}\mu}. \quad (15)$$

2) Calculation of the Bound for T: Let X be the random variable for the number of attacks success in the anonymous routing path. The random variable X has a binomial distribution, which can be obtained using p_m^d and it is given as:

$$P(X = x) = \binom{T}{x} \left(p_m^d\right)^x \left(1 - p_m^d\right)^{T-x}. \quad (16)$$

Thus, (16) shows the probability of x attacks success in T trials. Then $E[X] = Tp_m^d$ and (15) can be written as:

$$P[X < (1-\delta)Tp_m^d] < e^{-\frac{\delta^2}{2}Tp_m^d}. \quad (17)$$

Here we bring in ε (where $0 < {}^{\forall}\varepsilon < 1$) to obtain the bound of T. We can write the right side of (17) as

$$e^{-\frac{\delta^2}{2}Tp_m^d} < \varepsilon. \quad (18)$$

Then, by setting $\varepsilon = \frac{1}{\log n}$, we have

$$T > \frac{2\log(\log n)}{\delta^2 p_m^d}\log n. \quad (19)$$

Following (18), we can say that if T satisfies (19), then for the number of attacks X we have $X > (1-\delta)Tp_m^d$ with probability at least $1 - \frac{1}{\log n}$.

For the comparison of the information slicing method to the onion routing method, we define $\frac{X}{T}$ as *the attack rate* per round. When T satisfies (19), we have that $\frac{X}{T} > (1-\delta)p_m^d$. Additionally if we take T large enough, $\frac{X}{T}$ converges to $\mathrm{E}[\frac{X}{T}] = p_m^d$ by the law of large numbers. Now we analyze the lower bound of T when the attack rate $\frac{X}{T}$ exceeds certain threshold value ζ with probability more than, or equal to $1 - \frac{1}{\log n}$. In order to keep ζ constant, we set δ as $\delta = 1 - \frac{\zeta}{p_m^d}$.

If $\zeta < p_m^d$, δ is positive. In the calculation, we set ζ which satisfies $\zeta < p_m^d$, δ when p_m^d becomes smallest value.

Figure 8 gives a comparison of the number of rounds T that should be made in order that the attack rate $\frac{X}{T} > \zeta$ in case of the information slicing method and the onion routing method. Note that the case for $d = 2$ is the same as the onion routing method. In this calculation, we set $n = 1000$ and $\zeta = 1.0 \times 10^{-11}$. We can see in Figure 8 that the graphs for T in case of the information slicing method for different values of d are always above the graph for the onion routing method for $p_m = 0.1$, $p_m = 0.2$, and $p_m = 0.3$. It means that in order to make $\frac{X}{T} > \zeta$, much more rounds T should be made in case of the information slicing method compared to the onion routing. Thus, we can conclude that the anonymity of the information slicing method is much higher than that of the onion routing method for any d.

The comparison allows us to say that our proposed mutual anonymity protocol provides better anonymity than the conventional methods.

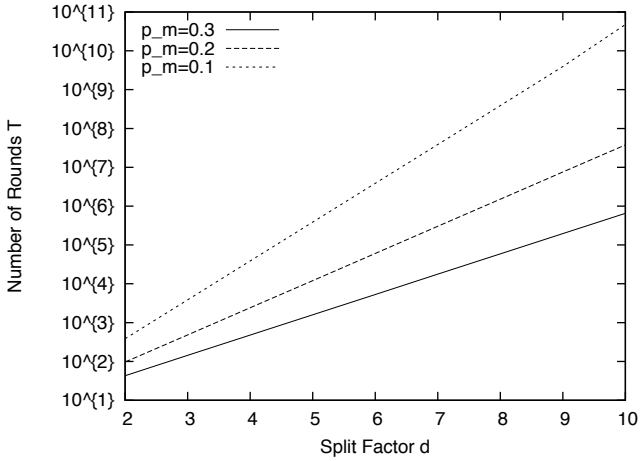

Fig. 8. Comparison with Bound of Round T

VII. Conclusion

In this paper a mutually anonymous protocol for decentralized P2P networks has been proposed. The protocol is a combination between the SSMP and the information slicing technique and guarantees initiator's anonymity, responder's anonymity and anonymous communication between them. Employing the concept of secret sharing schemes plays an essential role for the protection of the transmitted information between the initiator and responder, and using the information slicing technique the protocol is churn resilient and can be realized with lower cryptographic cost. Moreover, the anonymity evaluation shows that the proposed mutual anonymity protocol provides higher anonymity than the conventional methods.

As future works, we consider the practical implementation of the proposed protocol. It is essential to evaluate the performance of the protocol by using it in a real system. We consider to make experiments to measure the computational costs and communication overhead of the proposed protocol as well.

References

[1] J. Han, Y. Liu, L. Xiao, R. Xiao, and L. Ni, "A Mutual Anonymous Peer-to-peer Protocol Design," *Proc. of 19th International Parallel and Distributed Processing Symposium (IPDPS'2005)*, pp. 68, 2005.

[2] J. Han, Y. Zhu, Y. Liu, J. Cai, and L. Hu, "Provide Privacy for Mobile P2P Systems," *First International Workshop on Mobility in Peer-to-Peer Systems (MPPS) (ICDCSW'05)*, vol. 8, pp. 829-834, 2005.

[3] J. Han and Y. Liu, "Mutual Anonymity for Mobile Peer-to-Peer Systems," *IEEE Transactions on Parallel and Distributed Systems* , vol. 19, no. 8, pp.1009-1019, 2008.

[4] S. Katti, J. Cohen, and D. Katabi, "Information Slicing: Anonymity Using Unreliable Overlays," *Proc. of 4th USENIX Symposium on Networked Systems Design and Implementation*, pp.43-56, 2007.

[5] S. Katti, D. Katabi, and K. Puchala, "Slicing the Onion: Anonymous Routing without PKI ," *ACM HotNets*, 2005.

[6] A. Shamir, "How to Share a Secret," *Communications of the ACM 22,* pp. 612-613, 1979.

[7] P. Syverson, D. Goldschlag, and M. Reed, "Anonymous Connections and Onion Routing," *Proc. of IEEE Symposium on Security and Privacy,* 1997.

[8] M. Wright, M. Adler, B. Levine, and C. Shields, "The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems," *ACM Transactions on Information and Systems Security*, vol. 7, no. 4, pp. 489-522, 2004

[9] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, "Towards an Analysis of Onion Routing Security," *Workshop on Design Issues in Anonymity and Unobservability*, pp. 96-114, 2000.

[10] R. Nelson, "Probability, Stochastic Processes, and Queueing Theory," Springer-Verlag, 1995.