Xiao RY. Survey on anonymity in unstructured peer-to-peer systems. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(4): 660–671 July 2008

Survey on Anonymity in Unstructured Peer-to-Peer Systems

Ren-Yi Xiao (肖人毅)

National Natural Science Foundation of China, Beijing 100085, China

E-mail: xiaory@nsfc.gov.cn

Received December 22, 2007; revised April 10, 2008.

Abstract Although anonymizing Peer-to-Peer (P2P) networks often means extra cost in terms of transfer efficiency, many systems try to mask the identities of their users for privacy consideration. By comparison and analysis of existing approaches, we investigate the properties of unstructured P2P anonymity, and summarize current attack models on these designs. Most of these approaches are path-based, which require peers to pre-construct anonymous paths before transmission, thus suffering significant overhead and poor reliability. We also discuss the open problems in this field and propose several future research directions.

Keywords unstructured peer-to-peer systems, mutual anonymity, privacy, distributed system

1 Introduction

Recently, Peer-to-Peer computing (P2P) has become a promising solution to resource sharing in large scale distributed networks. Different from the traditional client/server model, P2P computing is able to aggregate and fully utilize resources from all users instead of a few central servers.

However, current P2P systems have not addressed an important issue: how to achieve user's anonymity[1−3]. The major anonymity concerned with P2P users is that the users' identities and actions can be revealed by other members. In current P2P systems, attackers may make use of some flaws, such as plain-text query, exposed IP address, and direct file-downloading, to compromise user anonymity. The open and distributed features of P2P systems make the situation of user anonymity even worse.

In this paper, we focus on the existing anonymous approaches in unstructured P2P systems. We propose a new taxonomy for identifying and describing those anonymous approaches. The taxonomy categorizes existing works based on the patterns of message delivery, such that the prime features of anonymous techniques are explicitly summarized and distinguished. This paper contributes a comprehensive survey on current studies. We also discuss and summarize the challenges in the anonymous P2P community.

The paper is organized as follows. Section 2 introduces the background knowledge of anonymity and P2P systems. Section 3 proposes the taxonomy and outlines existing approaches. We summarize major attacks targeted on anonymous P2P applications in Section 4. We discuss open issues in Section 5, and conclude the paper in Section 6.

2 Background and Fundamental Techniques

Privacy is a basic requirement of secure social behaviors such as voting, participating in surveys, reporting crimes and the like. It is critical to protect the above actions from prying "eyes" and the illegal surveillance. The motivation behind anonymity is to protect the private information, say privacy, for users. Anonymity has become a major method of protecting our privacy. ISO/IEC 15408-2[4] defines the anonymity as follows. "Anonymity ensures that a user may use a resource or service without disclosing the user's identity." According to this definition, the crucial function of anonymity is to protect users' identities. In terms of the computer community, the anonymity requirement is especially important for those users who want to protect their personal, private, and sensitive information, such as the user name, ID, and IP address, during communication with others.

Peer-to-Peer (P2P) is a creative model motivated by the requirement to share and cooperate in a large scale distributed system. Different from the traditional client/server model, the P2P model fully utilizes the resources of all nodes in the system instead of only a small number of central servers. The basic idea of a P2P model is to build a virtual layer over the applica-

tion layer or network layer. In such an overlay network, all hosts, called peers, interconnect with each other, and cooperate to perform computing tasks and share resources. Each peer in this overlay is both a server and a client, that is, all peers are both the resource consumers and providers. Currently, file sharing is the most popular application in P2P systems.

P2P systems can be divided into two types: structured P2P and unstructured P2P.

Structured P2P systems, such as Chord[5], Pastry[6], Tapestry[7], CAN[8], map each node as well as the index information of each resource into a position in a highly organized structure. The structure, usually a ring, is constructed by using a globally consistent scheme such as Distributed Hash Table (DHT). Because of the highly organized DHT, routing a query to desired resources in structured P2P systems is very efficient, even if the resource is rare. However, two main drawbacks limit the implementation of structured P2P. First, the structured P2P cannot support the fuzzy query since all queries should be hashed before issuance. Second, the construction and maintenance of the DHT structure introduce large overhead to individual peers. In addition, each peer has to store the index information of those resources which are belonging to other peers.

Unstructured P2P file sharing systems, such as Napster[9], KaZaA[10], BitTorrent[11], and Gnutella[12], are more popular. Peers are simply interconnected in an ad hoc pattern and there does not exist any structured pattern in these systems. In this survey, we will mainly focus on the unstructured P2P systems[13,14]. Unstructured P2P systems can be classified into three categories: centralized, decentralized, and hybrid. A centralized unstructured P2P system, such as Napster[9], holds one or more centralized servers to provide resource index services. Those servers maintain index lists of available resources of all peers. Each peer sends requests for desired resources, called queries, to the index servers. For each query, index servers search in the maintained index lists and reply a result to the requesting peer. The response includes the description of resources and providers' IP addresses. Upon responses, the requesting peer chooses a desired responder and directly contacts the responder to download the resource. Fig.1 depicts this procedure. Centralized P2P benefits from the efficient search performed by index servers. However, the overt drawback is that index servers are vulnerable to single point of failures and denial of service attacks.

As for decentralized unstructured P2P systems[13,15], they remove index servers and are widely deployed[16]. Instead of processing queries in a centralized manner, peers usually employ a flooding mechanism to issue queries. As the example shown in Fig.2, each requesting peer broadcasts a query to its neighboring peers. The query is broadcast and rebroadcast in the system, which is called a flooding procedure. Each peer caches a local routing table for relaying queries. If a peer within the flooding scope has a matched object, it becomes a responder. All responders deliver their responses to the requesting peer. Each response is delivered along the reversed path of the query message until it reaches the requesting peer. The initiator then selects a desired responder and directly downloads the resource from the chosen responder. To keep the flooding scalable, a TTL (Time-To-Live counter) value is set in the query message to constrain the hops it traverses. The decentralized model is more reliable than the centralized model due to the elimination of centralized servers. However, the flooding search incurs a large amount of traffic cost and degrades the search efficiency[17−19].

Combining the advantages of the centralized and decentralized models, the "hybrid" unstructured P2P model[20] improves the search efficiency while maintaining the reliability. The hybrid unstructured P2P comprises a larger number of small groups, as shown in Fig.3. Each group is a small centralized P2P system,

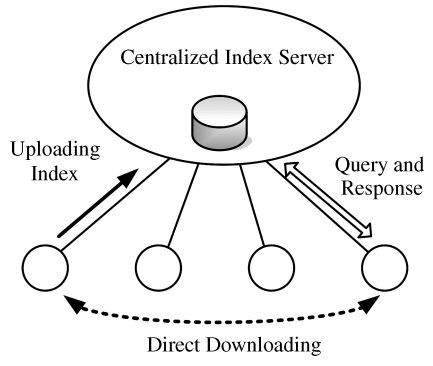

Fig.1. Centralized P2P.

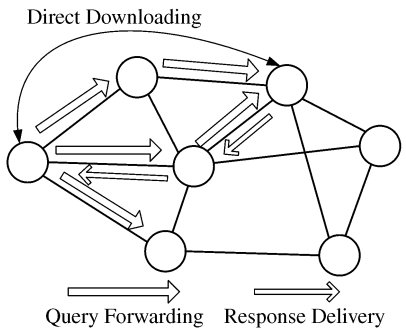

Fig.2. Decentralized P2P.

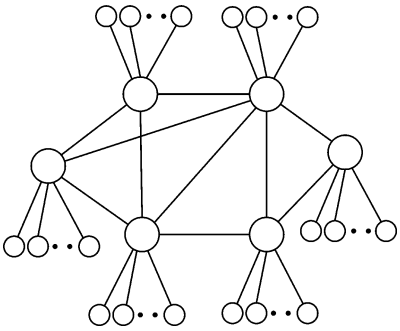

Fig.3. Hybrid P2P.

with a group leader, called super peer, behaving as the index server for other group members. All super peers are organized into a decentralized overlay. The most successful application in this category is KaZaA[10]. The same adoption is also recently seen in Gnutella.

Anonymity in P2P systems includes publishing anonymity, sending anonymity (initiator anonymity), and receiving anonymity (responder anonymity). Publishing anonymity usually means that users create something without being discovered. Such a requirement is also called the censorship-resistance, which is mostly concerned with free resource sharing systems. The sender/receiver anonymity protects senders/receivers from being exposed to other entities during the message delivery. These two requirements usually merge together to provide a complete anonymity. In addition, to protect the transferred data, cryptographic operations are performed in the message delivery. Thus, a complete anonymity comprises sending anonymity, receiving anonymity, and secure transmissions among participants. We call it a mutual anonymity.

The designs of unstructured P2P models do not provide complete anonymity solutions to users. First, without any cryptographic operations, peer identities are exposed to their neighbors during communications. The peer anonymity will be compromised when there are eavesdropping neighbors. In addition, attackers also make use of some control information of the packets to locate peers. For example, Gnutella uses a TTL counter, initially 7 hops, to limit the flooding scope in query packets. The value in this counter will be subtracted by one after each relaying until it becomes zero. When an attacker receives a query with 6 hops, it can immediately deduce that the upstream node must be an initiator.

To enhance the users' privacy, existing anonymous approaches have adopted cryptographic techniques in P2P systems, including random number, hash function, cipher, multicast and broadcast, and secret sharing. Those techniques are employed to provide the confidentiality, message integrity, and anonymous delivery.

3 Taxonomy of Unstructured P2P Anonymous Approaches

Previous taxonomies[21−23] mainly focus on the method of choosing anonymous agents or the routing pattern. Although choosing anonymous agents is important in the construction of anonymous channels, existing taxonomies are coarsely granular and not comprehensive. As we mentioned before, the ultimate goal of anonymous P2P applications is to hide the user identities, such as the user's ID and IP address. To accomplish this purpose, researchers focus on anonymizing the message transmission, since the communications among users are completed via the message delivery. In fact, anonymity can be regarded as a special encryption on the messages to conceal correlations between the messages and the senders. The anonymizing process is performed during publishing, communicating, searching, and retrieving. Therefore, protecting the messages in communication is essential for anonymity. We propose a new taxonomy based on the treatment pattern to messages in the anonymizing procedure. Our taxonomy provides an insight, comprehensive, and finely granular investigation on existing anonymous approaches.

On the basis of our taxonomy, the existing anonymous approaches in unstructured P2P systems can be divided into three categories: unimessage-based, split message-based, and replicated message-based. Briefly speaking, a unimessage-based approach delivers each message as a single one. Thus, the number of messages will not be changed. Split message-based approaches, however, divide each message to fragments, and the receiver can only recover the original message by collecting enough fragments. Replicated message-based approaches, however, replicate each message to multiple copies and spread them in the system.

3.1 Unimessage-Based Approach

Most anonymous approaches are belonging to this category. They achieve anonymity by subtly encrypting messages and assigning a single anonymous path for the message delivery. The term "unimessage-based" comes from the observation that a message is handled as an entire packet during the anonymous communication in those approaches. The main objective behind unimessage-based approaches is to hide the path. Works in this category are also called path-based approaches. They pre-construct anonymous paths before transmission. The basic versions of unimessage-based approaches are Mix or Onion Routing, as illustrated in Fig.4. In this example, the sender I transmits data through a path: $I \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow R$. The path is hidden, using a layer-encrypted data structure. The objective of this layer-encrypted data structure is that each node in the path only knows its successor's IP address and has no knowledge about I's IP address and the content of the message. R can recover the data, but does not know I's IP address. I organizes the packet in the following way. The innermost layer includes the IP address of R, the receiver, and the original data encrypted using R's public key. I then wraps this layer by encrypting it using the public key of R's predecessor. Along

the reversed sequence of the nodes in the path, I keeps wrapping the packet in such a layer-encrypted pattern until it reaches the first intermediate node, node 1 in the example in Fig.4. Thus, each intermediate node in the path is related to a layer encrypted using its public key in the packet. The layer includes the IP address of this intermediate's successor and an encrypted inner layer. During transmission, each intermediate node decrypts the received packet using its own private key. It then gets an inner layer and the successor's IP address. To the intermediate node, the inner layer is a garbage because this layer is encrypted using the successor's public key. The only thing this intermediate node can do is to forward the inner layer payload to the successor. This procedure continues until the innermost layer reaches the destination node. In this way, the path is hidden from all nodes, including R and all intermediate nodes, except I. I hereby achieves the sender anonymity in this procedure. Unimessage-based approaches do not replicate messages so that message delivery incurs low traffic overhead.

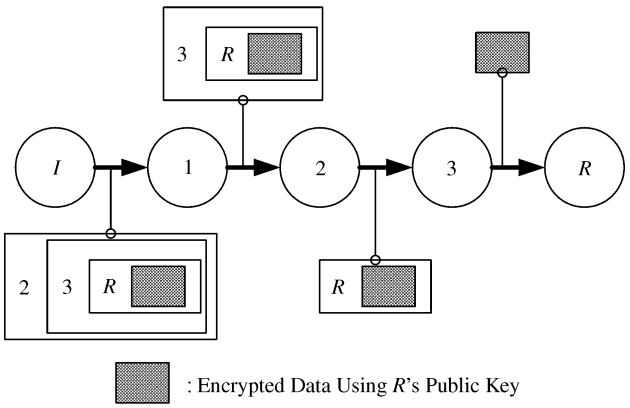

Fig.4. Mix and Onion Routing.

The unimessage-based approaches can be further subdivided into three groups: fundamental path-based, probability-based, and mimic traffic-enhanced approaches. Works in the first group achieve the anonymous transmission by directly implementing the basic Mix or Onion Routing scheme to construct anonymous paths. To enhance the unlinkability of forwarding, the approaches in the second group perform a probability-based delivery mechanism along the anonymous paths. The third group includes those approaches which employ mimic traffics in the anonymous transmission to further obscure the observers.

3.1.1 Fundamental Path-Based Approach

Fundamental path-based approaches employ Mix or Onion Routing technique to build the anonymous system. We outline the essential features of the fundamental path-based approaches by discussing two representative protocols: APFS[24] and Tor[25].

APFS[24] deploys a bootstrapping node called coordinator to deal with the anonymous path construction. This node must be always online. It provides a list of online peers to fresh peers for constructing anonymous paths. Each peer constructs an onion path pointing to another peer, called tail node, which acts as an anonymous transferring agent. Some peers volunteer to become servers to index the resources of the system. Server peers post their tail nodes to the coordinator. Thus, the client peers can upload their resource lists and requests to the server peers through onion paths. The response messages and the target file are also delivered through the onion paths. Therefore, APFS provides anonymous file retrieval service to the P2P systems. APFS also performs a multicast as a replacement for the coordinator to strengthen the reliability.

Tor[25] is an advanced version of Onion Routing. Instead of using a single layered encryption packet, say an onion, Tor implements an incremental path-construction in which the initiator extends the path hop by hop and negotiates session keys with each intermediate node on the path. As a benefit, the anonymous transmission is more reliable since the intermediate nodes on the path are online after the path construction. Tor is more convenient than Onion Routing in supporting TCP-based applications.

Other researches adopting the fundamental path-based idea include: MorphMix[26] and GAP[27]. MorphMix[26] focuses on enlarging the anonymous proxy group to improve the anonymity. Each MorphMix client is a Mix node in the system so that the anonymous proxy group is extended to the entire system. As a result, the anonymity set, which includes all possible initiators, is the entire system. This design augments the difficulty in guessing the initiator identity since the number of suspected nodes is maximized from the observers' perspective. GAP[27] allows censorship-resistant file-sharing over a Mix-based network.

3.1.2 Probability-Based Approach

A number of path-based approaches such as Crowds[28], Shortcut[29], and AP3[30] allow the intermediate nodes on anonymous paths to perform a probabilistic forwarding to strengthen user anonymity. We define them as probability-based approaches.

Crowds[28] is an anonymous web transaction protocol. Crowds only provides sender anonymity. In Crowds, each intermediate node, called *jondo*,

randomly chooses a successor to forward the request, or directly deliver the message to the destination. Each link in Crowds is protected by performing a symmetric cipher. This mechanism is widely used for achieving sender anonymity. The key negotiation and distribution among peers increase the complexity of Crowds and incurs much overhead.

Shortcut[29] outperforms existing unimessage-based forwarding works in reducing the communication latency and response time. In this protocol, peers still achieve anonymity via onion paths. For anonymous replying, an initiator establishes an onion-based reply block, called re-mailer, and encapsulates it into the query. Such a re-mailer is an anonymous return path. Then the query is probabilistically forwarded in the system. Each peer that receives the query either acts as a reply agent or continues forwarding the query based on a probability. If a peer acts as the reply agent for the initiator, it adds its IP address in the query message and forwards this message to a randomly chosen neighbor. From this point, each node receiving the packet either forwards the packet to a randomly chosen node, or floods this query into the system, based on a probability. Upon a request, a responder builds an onion path to anonymously send the file to the reply agent. The file is then delivered along the re-mailer until it reaches the initiator. The key contribution of this work is that the length of the response path is usually much shorter than that of the requesting path by using the re-mailer.

*AP*3 [30] is similar to Crowds but operates on top of application layer. In AP3, each node can be mapped to a key, which is a kind of coordinates. An intermediate node flips a weighted coin to decide whether it sends the received message to the intended recipient. If not, this node chooses a random key and delivers the message to the neighbor that is "closest" to the key. In AP3, all nodes keep a local routing table to cache the message route information so that the response can be delivered via the reversed path of the anonymous channel.

3.1.3 Mimic Traffic-Enhanced Approach

Another improvement to the fundamental path-based technique is to introduce mimic traffic to the system. The mimic traffic can help hide the data flows in P2P overlay such that it is difficult to distinguish the real flow from those noisy ones.

Tarzan[31] provides a best-effort delivery service over IP layer. Each node of Tarzan is based on fundamental path-based technique to anonymously deliver messages. Different from onion routing which only provides a small proxy set, each Tarzan peer involves all other nodes in its proxy set. To accomplish this, Tarzan uses a gossip-based protocol for proxy discovery. The most elegant design in Tarzan is to inject mimic traffics to communication links to protect real data flows against eavesdropping. In Tarzan's topology, each node establishes k bidirectional links with k neighbors. All nodes maintain and balance the mimic traffics according to a number of criteria to shape the traffic into a time-invariant pattern. This defends the real traffic against being distinguished from the mimic ones. However, Tarzan's architecture is insufficient to guarantee a rapid flux in P2P systems. Tarzan's proxy discovery scheme and key exchange mechanism also incur significant amount of traffic.

D. Liu *et al.* proposed a procedure to normalize the traffic pattern with anonymity concerns[32]. Their main strategy of shaping the dummy traffic is similar to Tarzan. The basic idea is also to send both real and mimic packets over encrypted links with constant size and interval between them. In addition, they also simply generate each mimic packet by splitting the last real packet into two fragments and permuting them as a new packet. The splitting position of the last real packet is randomly chosen. This method improves the efficiency of generating mimic traffic. However, it is not secure as the way to encapsulate the entire mimic packet with random bits.

The mimic traffic mechanism[31−33] significantly improves the anonymity for users. However, the mimic traffic inevitably incurs a large amount of traffic overhead.

Note that our taxonomy method for unimessage-based approaches is based on their most characteristic features in the anonymity achievement. Some approaches are in the interaction of above three subgroups. For example, Tarzan employs both the basic Onion Routing and mimic traffic to enhance the anonymity.

In summary, unimessage-based approaches incur the lowest traffic overhead. They also provide a high anonymity guarantee because both the messages and message delivery paths are encrypted. They, however, suffer a number of drawbacks. First, an initiating peer must obtain enough proxies to pre-construct anonymous paths, which is reluctant especially for those fresh peers. Second, the anonymous paths are not reliable and difficult to maintain. An initiator is not aware of the availability of the chosen proxies. Because of the high dynamic nature, P2P systems cannot guarantee that each chosen proxy on the paths is active during the message transmission. Finally, the computational cost

of cryptographic operations is high due to hop-by-hop asymmetric key-based encryption/decryption mechanism.

3.2 Split Message-Based Approach

Split message-based approaches employ secret sharing scheme to achieve anonymity. The secret sharing scheme splits a secret into several fragments, called shares, and distributes them to individual users. A threshold is set for secret reconstruction. The number of collected shares must be equal to or larger than that of the thresholds such that the secret can be recovered. In P2P systems, secret sharing can be employed to achieve publishing anonymity. We define those anonymous approaches are split message-based if they utilize the secret sharing like schemes to divide a message into multiple fragments to achieve user anonymity.

Free Haven[34] is an anonymous publication and storage system providing a censorship-resistance for users. Free Haven employs Information Dispersal Algorithm (IDA), a kind of secret sharing scheme, to break the file into fragments. A single fragment will not disclose the publisher ID and content of the file. The anonymous communication is constructed via Mix technique. A community of servers, known as a servnet in Free Haven, host and exchange fragments with others. When a provider publishes a document, it splits files by using IDA, sets a threshold for file reconstruction, marks the fragments with a unique ID, and uploads the shares to one of the servnets. Servnets publish received shares and exchange some fragments with other servnets. A requester issues a query containing the ID of desired file to any servnet. The servnet floods the request and receives the fragments replied from responding servnets. The fragments are delivered via Mix-based anonymous paths. Free Haven infrastructure provides a reliable data retrieval for users via the redundant fragments. As a tradeoff, both the fragments trading procedure and fragments storage may incur significant overheads to the system.

SSMP[35] provides mutual anonymity to unstructured P2P systems. The authors suggested to perform the secret sharing scheme in the query issuance and file downloading. In SSMP, instead of directly issuing a query, an initiator splits the query into shares and sends them to a number of neighbors. The shares are then flooded in the system. To keep the traffic overhead caused by share flooding, SSMP employs a probability-based flooding, in which each intermediate node either sends a share to a randomly chosen neighbor or broadcasts the share. The decision making is based on a probability. Once a node collects enough shares, it can recover the query and flood this query to the system. SSMP also adopts the similar idea on the file delivery. The cryptographic computation overhead is relatively small because the secret sharing scheme causes a smaller computation overhead than the asymmetric key-based ciphers such as RSA (proposed by Ron Rivest, Adi Shamir and Leonard Adleman). However, this work still suffers a large traffic overhead caused by share flooding. PUZZLE[36], which extends SSMP to mobile P2Ps, mitigates this problem because the flooding is a fundamental communication pattern in mobile environments.

RR[37] is a mutual anonymity protocol aimed at reducing the overhead as well as improving the reliability of anonymous P2P systems. Rumor Riding (RR) protocol is based on a random walk scheme. RR does not need to construct asymmetric cipher encrypted anonymous paths such that it greatly decreases overhead in anonymous communications. RR employs AES, a symmetric key cipher to perform encryption. Instead of only sending the cipher like path-based approaches, RR drives both the cipher and key of a message randomly walking in the P2P systems. The cipher and key are called cipher rumor and key rumor, respectively. RR allows each peer to adaptively determine the length of rumors to guarantee a pair of rumors to meet with a high probability. The peer that receives a pair of rumors can recover the original query. This peer then floods the query on behalf of the unknown initiating peer. Similarly, RR achieves anonymity for the response, confirmation, and file-downloading procedures in a query cycle. The highlights of RR include: low cryptographic overhead due to the usage of symmetric key cipher, low traffic overhead due to small amount of rumors compared to secret sharing-based approaches, reliable anonymous communication, and high anonymity guarantee. Since all the intermediate nodes randomly choose the successor for a received rumor, different rumors would be delivered along different paths. In contrast, anonymous paths are fixed in most of unimessage-based approaches. Thus, some attack methods which are effective to unimessage-based approaches would be infeasible to RR.

The splitting message-based approaches[35,37−40] provide a promising direction in solving the anonymity problem. Users may assign arbitrary anonymous sets and deliver messages by implementing the secret sharing distribution or random walk mechanism. However, the efficiency is still a challenging issue in this category.

3.3 Replicated Message-Based Approach

In this category, researchers implement broadcast

and multicast to achieve anonymity. An example might illustrate it well. Suppose two partners want to communicate anonymously with each other in an open environment. They can speak loudly in such a jargon that only they can understand the meaning of the sentences. Thus, even though other people within the sound scope can hear the talking, they cannot understand the content of the conversation.

To hide the content of the message, users usually perform encryption to the messages using the receiver's public key. Thus, only desired receivers can recover the original data. In such a procedure, we find that the messages are propagated into replicate copies, and usually delivered by broadcasting or multicasting.

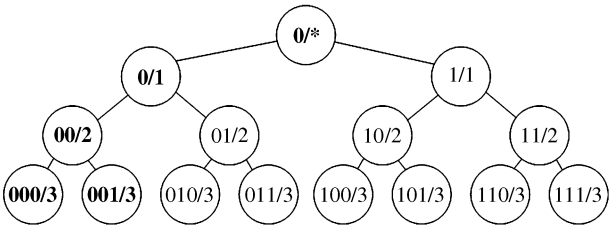

Fig.5. Logical binary tree of P^5.

P^5[41] is the representative work in this category. In P^5, all participants in the same broadcasting group, termed as a "channel" in this paper, send fixed-length packets at a fixed rate. In this channel, each message is encrypted using the receiver's public key and is broadcasted to all the other peers. Although all the peers can receive this message, only the receivers can recover the message. The receivers, however, do not know the senders' identities since it is possible that any other peer can send this message to them. To disabled attackers from traffic analysis, P^5 introduces noise packets to keep a fixed transmitting rate for each user. P^5 also designs a clever hierarchical binary tree to partition all users into different broadcasting groups, as shown in Fig.5. In this binary tree, each logic node represents a broadcast group, say a channel. Each group ID consists of two items: a bit-string and a mask. The mask means the bit number of the corresponding bit-string. When a user joins the system, it first calculates a hash value of his public key, for example $001 \cdots 010$. Then it randomly selects a mask (2 in Fig.5), and compares the first sub-bit-string of its hash value of the public key (the users' sub-string is 00 in Fig.5). It can uniquely locate a group with a matched ID (00/2 for the example in Fig.5). Therefore, each peer can join a number of groups in this way. The broadcasting rules are as follows. Any message sent to a group is forwarded to all members of this group, all groups in its subtree, and all upstream groups tracing back to the root. On the basis of this design, users can choose the scope of broadcasting on a tradeoff between anonymity and the communication efficiency. However, a sender only knows to which group the receiver may be belonging, but has no idea which specific group the receiver really stays. Due to the elegant design of channels and hierarchy overlaid spanning trees, P^5 genuinely reduces some of the traffic caused by broadcasting.

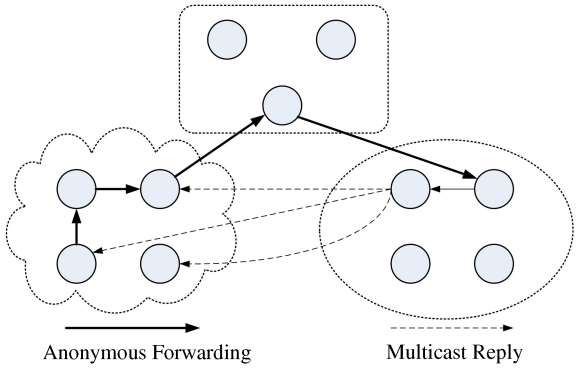

Fig.6. Hordes' infrastructure.

Hordes[42] provides sender anonymity by adopting the Crowds probabilistic forwarding mechanism, and achieves receiver anonymity by performing a multicast transmission. Fig.6 presents the main infrastructure of Hordes. Since the replying path is the shortest multicast path from the responder to the initiator, Hordes significantly reduces the response time. However, peers in Hordes must participate in the multicast relaying, which incurs a huge traffic and wastes the bandwidth.

Other replicate message-based approaches[43,44] also need some kinds of broadcast or multicast channels. However, the utility of this kind of work is rigorously constrained due to the huge number of redundant messages caused by the broadcast and multicast. Meanwhile, since each node has to decrypt all the received packets, the cryptographic overhead is accordingly tremendous.

3.4 Anonymity in Structured P2P Systems

Besides the unstructured P2P systems, structured P2P systems[6−9,45] also gain much concern. The structured P2P systems usually employ a Distributed Hash Table (DHT) to build a location-mapping structure. The system maps each node and file to a specific position in the DHT-based structure, and performs exponentially incremental routing algorithm to search the desired items. Searching a given item in such a structure needs $\log(n)$ steps, where n is the number of nodes

Table 1. Comparison Between Representative Approaches

Approaches	Type	Anonymity	Overhead	Reliability	Efficiency
APFS	UniMsg(Fun)	High	Medium	Low	Medium
Tor	UniMsg(Fun)	High	Medium	Low	Medium
Crowds	UniMsg(Pro)	Medium	Low	Low	Medium
Shortcut	UniMsg(Pro)	High	Low	Low	Medium
Tarzen	UniMsg(Mim)	High	High	Medium	Low
Free Haven	SplMsg	High	High	High	Low
SSMP	SplMsg	Medium	High	High	Low
P^5	RepMsg	High(Adaptive)	Very High	High	Low
Hordes	RepMsg	High	Very High	High	Low

Note: UniMsg: unimessage-based; Fun: fundamental path-based; Pro: probability-based; Mim: mimic traffic enhanced; Splmsg: split message-based; RepMsg: replicate message-based.

in the system. This is the most outstanding feature of structured P2P systems. DHT gives latency bounds of the search procedure in distributed networks.

Unfortunately, structured P2P systems are not practical. The main reason is that nodes are uniformly distributed over the DHT structure, while in practice this logic structure does not match the underlying network. In terms of anonymity, the structured P2P systems have to pay more efforts to achieve the same anonymity than in unstructured systems. Since DHT structure is widely-known, adversaries can pre-calculate the position where the content should be published. In addition, the mapping mechanism in structured P2P systems allows adversaries to easily locate each node's position in the structure. Thus, the anonymity in structured P2P is at least as the same challenging as in the unstructured P2P systems. On the other hand, anonymity also incurs extra overhead in structured P2P systems. Existing anonymous approaches[46,47] in structured P2P systems do not outperform those in unstructured systems, either in terms of the anonymity degree or the system performance. Since we just focus on the unstructured P2P systems, the deeper discussion about the anonymity in structured P2P is out of our scope.

4 Attacking Attempts in Anonymous Systems

Anonymous P2P applications also suffer different attacks. The attackers may be system members or intruders from outside. The ultimate target is to locate the sender, receiver, and find what they are transferring. Individual attackers may collaborate to make their attacks more effective. For attackers, the topology of the P2P system is very important, because the overlay topology provides the attacker with necessary knowledge about the system connection and data transfer, and having a closer distance between attackers and victims reduces the difficulty of detection.

Attackers' behaviors can be classified as follows.

Time-to-Live Attacks: time-to-live (TTL) counters determine the maximum number of hops for a message to traverse. The TTL value of a message is decremented at each node in the P2P systems to prevent infinite delivery. As mentioned previously, attackers may make use of this information to locate the message sender. The detection is more effective when the TTL value is small. For example, in Gnutella, if a malicious node receives a query message with a TTL value as 6 (the default TTL value is set to 7), the node can immediately deduce a conclusion that its predecessor of the query message is the initiator. In contrast, this method might not work if the system allows a changeable TTL value. Another effective protection is to enlarge the TTL value. From the attackers' view, a message with a large TTL value indicates the message has been relayed many hops. Tracing back to the initiator along such a "long path" is very difficult for attackers unless they own the global knowledge of system communications. Note the TTL attacking method is simple but very effective in non-anonymized P2P systems.

Cumulative and Statistical Attacks: any attacker can collect statistical data over a long time. When a sender repeatedly communicates with a receiver via fixed anonymous channels in a long period, the frequency of the initiator's appearances may expose their identities. Attackers accumulate knowledge obtained from the observation to locate the participants. The cumulative and statistical attacks are the main threats against the anonymous P2P systems. So far, a large number of prior approaches are based on this method.

The time and traffic analysis are two representative attacking models. Attackers may observe two suspected

flows in the system and check the correlation between them. The interval between two successive packets and the latency of packets are all such information to be used for analysis of whether two flows are belonging to one anonymous transmission. Similarly, attackers can detect traffic information among the suspect objects. Especially, attackers may intentionally clog or delay the flows to impose some shaping traffic if they are exactly in the transferring paths. Therefore, the adversaries can easily locate the identities of participants based on those correlations.

More specifically, Wright *et al.*[48] proposed a predecessor attack and investigate how effective this attack is when attacking anonymous system. The authors presented a series of upper bounds of rounds required for significantly degrading users' anonymity in current anonymous systems. V. Shmatikov proposed a probabilistic model checker, PRISM, to degrade the user anonymity in Crowds[49]. An interesting result is that a large Crowds system may help the attacker to locate the initiator. Y. Zhu *et al.*[50] proposed a model to measure the anonymity degree in terms of entropy, and report the relationship between anonymity degree and the capacity of anonymity-based covert channels. Y. Guan *et al.*[51] also gave a quantitative analysis to anonymous communication systems. They show that longer or complicated paths are better than shorter or simple ones and increasing the number of compromised nodes may cause the anonymity degradation. Note above two works are based on general anonymous systems. B. N. Levine *et al.*[52] presented a technique to thwart timing attacks in Mix-based systems. A. Serjantov and P. Sewell[53] investigated the time analysis for connection-based systems, e.g., Mix-based systems. X. Fu *et al.*[54,55] proposed several attacking models based on traffic analysis, which are effective to thwart the anonymity in both wired and wireless systems. Especially, the authors in these works point out that the traffic analysis attack may still potentially compromise the anonymity even though the padding method or mimic traffic is adopted. There are also a number of researchers[56,57] focusing on the traffic analysis attacks on real-time anonymous communication networks, for example, Tor, which can also be used in P2P systems.

To defend against such attacks, previous work usually keeps the probabilities of all interactive behaviors to be equal likely. However, adversaries may perform active attacks to the anonymous transmission channels such as modifying or dropping packets so that the receivers have to re-establish a connection. In this case, attackers may also be able to gain extra information they need for statistical analyses.

Research in attacks discussed above still stays in a preliminary phase. To our knowledge, most attacks to anonymous P2P applications derive from previous attacks to the Internet applications such as anonymous Email and Browser. Current attacks to anonymous P2P systems are not effective and are far away from practical deployments. First, most anonymous P2P systems have not been widely deployed yet. Second, the most important problem of those attacks in P2P systems is the difficulty of the attacker's observation. In current anonymous studies, a strong assumption for the attackers is that they can obtain a global knowledge of the whole system. This assumption is set in this way so that researchers can calculate the low bound of the user anonymity degree in those approaches. However, it is almost impractical or extremely difficult for attackers in real P2P systems, which lack any centralized mechanism, to collect such global information. Even if the attackers can achieve a kind of global information, the data collection and computation cost is extremely high in large scale P2P systems. Third, the evaluation of attack efforts cannot guarantee a high accuracy. The reason is similar to the evaluation of the anonymity degree, which will be discussed in the next section. Last, the data reflecting attacking results is very difficult to collect. So far, the log data and record for any attacks on the anonymity are scarcely seen in unstructured P2P systems. To further investigate the properties and efforts of those attacks, researchers are encouraged to implement those methods in real systems and make more comprehensive analysis.

5 Open Research Issues

There are a number of challenging issues in anonymizing P2P systems. First, anonymity is still a largely unexplored area in P2P systems. Users wish for more secure anonymous applications; government and copyright organizations require powerful and effective surveillance tools to prevent against illegal usage of anonymous P2P systems; and service providers focus on the impact of the system reliability if performing anonymous applications. The ongoing P2P anonymous work should be designed to deal with the above requirements. Unfortunately, these requirements conflict with each other. Balancing the above requirements needs a well defined anonymity model and accurate measurement for the degree of anonymity. Beyond these, it is further concerned that how anonymity services coordinate with other P2P system functions such as the trust management and incentive mechanism. In the following we overview the most heatedly discussed problems to sketch the main challenges in the anonymous P2P

approaches.

• *Anonymity Evaluation.* As we mentioned before, users need to evaluate their degrees of anonymity provided by anonymous applications. However, it is extremely difficult to construct a comprehensive model for evaluation of the degree of anonymity in unstructured P2P systems. The reasons include the heterogeneity and the highly dynamic topology of the P2P system. The heterogeneity features can be found in most famous P2P systems. For instance, the link degree of peers follows the power law relationship; the P2P systems usually have a small world phenomenon; and the resource popularity exhibits a Zipf distribution[58]. Meanwhile, current hybrid P2P systems also show a dense connection among their super nodes. Currently, most popular P2P systems hold nearly millions or tens of millions online users. In such a huge large-scale system, the distribution and properties of nodes, links, and resources are too complicated to be easily characterized. Therefore, the evaluation of anonymity degree is challenging to researchers and should be explored in unstructured P2P systems according to the different models and architectures.

• *Tradeoff Between Performance and Anonymity.* The current anonymity approaches incur extra overhead to both the system and the participants. The overhead is caused by encryptions and decryptions, anonymous transmissions, and mimic traffics. Most approaches in unstructured P2P systems suffer the inefficiency problem. On the other hand, tons of researches have been proposed to improve the performance of the non-anonymous P2P systems by refining the overlay topology, resource localization, and data delivery. However, most of these studies need the user identities in their optimization, which would compromise the anonymity. For anonymous P2P applications, without the help of identities, such as IP addresses, the performance may be severely degraded. Thus, any development of anonymous applications has a challenging effect on the performance of P2P systems. When providing anonymity, new efficient anonymous solutions will be urgently required in future P2P research.

• *Surveillance and Traceable Measures.* P2P systems provide an open environment in which users can access the free resources provided by other unknown members. On the other hand, more and more copyright-protected productions suffer the illegal downloading via P2P file sharing. It is really a challenging issue to accurately detect or track illegal file trading or exchange especially in an anonymous system. Besides enforcing copyright laws and legislations, techniques for detecting, preventing, and tracking must be developed to constrain illegal file uploader's and downloader's behaviors in P2P systems. We have discussed the main attacking methods in Section 4. In fact, it is important to further study those attacking methods for fighting against the illegal downloading. From the technical viewpoint, future surveillance approaches should be effective to identify illegal users and gather evidence of their illegal activities. In particular, researchers should also prevent those attacks from being abused to compromise the anonymity of legitimate users.

• *Trust and Trust Management.* Currently, the conflict is irreconcilable between anonymity and trust. Most prior approaches of trust and trust management are identity-based, which means real user identities are needed to make authentication and verification. However, this mechanism does not work when considering user's anonymity. Even though many anonymous schemes correlate a real ID with a pseudonym, the trust problem becomes more difficult in the proof of the correlation between these two entities. Therefore, trust management schemes need to be further explored in anonymous P2P environments. Combined with anonymity, trust management systems should deal with the issues including the authentication[59], verification, reputation or credit record storage, auditing, and misbehavior reporting[60] in anonymous environments.

• *Incentive Versus Free-Riding.* A free-rider always consumes resources from others while contributing a little or nothing to the system. Because of the open nature of P2P models, the free-riding phenomenon is popular and degrades the system performance. Anonymity may exacerbate this problem since the free-riders cannot be located and selfish behaviors might be prevalent without any punishment. Most existing incentive schemes are implemented in non-anonymous environments[61,62]. In those approaches, the awards and punishments are related to users' real identities. Therefore, it is difficult to build an incentive mechanism in anonymous environments. Researchers are encouraged to design incentive schemes that can effectively correlate the awards and punishments to certain users without destroying their anonymity.

6 Conclusion

To our knowledge, the approaches discussed in this paper are either in the theoretical stage or impractical to be implemented in real P2P applications. The tradeoff between efficiency and anonymity has not been well balanced. Some challenging issues are to be addressed. We surveyed the main research results on the anonymous P2P computing, and summarized several promising directions to address the existing problems. We

believe that, with the increasing effort spent on this topic, anonymous computing will become practical in P2P applications in the near future.

References

[1] Anonymity. http://freehaven.net/anonbib/topic.html.

[2] ISO IS 15408. 1999, http://www.commoncriteria.org.

[3] Pfitzmann A, Hansen M. Anonymity, unlinkability, unobservability, pseudonymity, and identity management — A consolidated proposal for terminology. Technical Report, 2005.

[4] ISO/IEC 15408-2. http://standards.iso.org/ittf/PubliclyAvailableStandards/, 2005.

[5] Stoica I, Morris R, Karger D, Kaashoek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. ACM SIGCOMM*, San Diego, California, USA, 2001, pp.149–160.

[6] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. Middleware*, Heidelberg, Germany, Nov. 2001, pp.329–350.

[7] Zhao B Y, Huang L, Stribling J, Rhea S C, Joseph A D, Kubiatowicz J D. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications* (*JSAC*), 2004, 22(1): 41–53.

[8] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. In *Proc. ACM SIGCOMM*, San Diego, California, USA, 2001, pp.161–172.

[9] Napster. http://www.napster.com.

[10] KaZaA. http://www.kazaa.com.

[11] BitTorrent. http://www.bittorrent.com/.

[12] Gnutella. http://gnutella.wego.com/.

[13] Liu Y, Xiao L, Liu X, Ni L M, Zhang X. Location awareness in unstructured peer-to-peer systems. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2005, 16(2): 163–174.

[14] Liu X, Liu Y, Xiao L. Improving query response delivery quality in peer-to-peer systems. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2006, 17(11): 1335–1347.

[15] Xiao L, Liu Y, Ni L M. Improving unstructured peer-to-peer systems by adaptive connection establishment. *IEEE Transactions on Computers*, 2005, 54(9): 1091–1103.

[16] Liao X, Jin H, Liu Y, Ni L M. Scalable live streaming service based on inter-overlay optimization. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2007, 18: 1663–1674.

[17] Liu Y, Zhuang Z, Xiao L, Ni L M. A distributed approach to solving overlay mismatch problem. In *Proc. the 24th International Conference on Distributed Computing Systems* (*ICDCS*), Hachioji, Tokyo, Japan, 2004, pp.132–139.

[18] Liu Y, Xiao L, Ni L M. Building a scalable bipartite P2P overlay network. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2007, 18(9): 1296–1306.

[19] Wang C, Xiao L, Liu Y, Zheng P. DiCAS: An efficient distributed caching mechanism for P2P systems. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2006, 17(10): 1097–1109.

[20] Xiao L, Zhuang Z, Liu Y. Dynamic layer management in super-peer architectures. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2005, 16(11): 1078–1091.

[21] Chothia T, Chatzikokolakis K. A survey of anonymous peer-to-peer file-sharing. In *Proc. IFIP International Symposium on Network-Centric Ubiquitous Systems* (*NCUS*), Nagasaki, Japan, 2005, pp.744–755.

[22] Rogers M, Bhatti S. How to disappear completely: A survey of private peer-to-peer networks. In *Proc. International Workshop on Sustaining Privacy in Autonomous Collaborative Environments* (*SPACE*), Moncton, New Brunswick, Canada, 2007.

[23] Nambiar A, Wright M. Salsa: A structured approach to large-scale anonymity. In *Proc. ACM CCS,* Alexandria, VA, USA, 2006, pp.17–26.

[24] Scarlata V, Levine B N, Shields C. Responder anonymity and anonymous peer-to-peer file sharing. In *Proc. the 9th International Conference of Network Protocol* (*ICNP*), Riverside, CA, USA, 2001, pp.272–280.

[25] Dingledine R, Mathewson N, Syverson P. Tor: The second-generation onion router. In *Proc. the 13th USENIX Security Symposium,* San Diego, CA, USA, 2004, pp.303–320.

[26] Rennhard, Plattner B. Introducing MorphMix: Peer-to-peer based anonymous Internet usage with collusion detection. In *Proc. ACM Workshop on Privacy in the Electronic Society,* Washington DC, USA, 2002, pp.91–102.

[27] Bennett K, Grothoff C. GAP — Practical anonymous networking. In *Proc. Privacy Enhancing Technologies Workshop,* Germany, 2003, pp.141–160.

[28] Reiter M K, Rubin A D. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1998, 1(1): 66–92.

[29] Xiao L, Xu Z, Zhang X. Low-cost and reliable mutual anonymity protocols in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2003, 14(9): 829–840.

[30] Mislove A, Oberoi G, Post A, Reis C, Druschel P, Wallach D S. AP3: Cooperative, decentralized anonymous communication. In *Proc. the 11th ACM SIGOPS European Workshop,* Leuven, Belgium, 2004, Article No.30.

[31] Freedman M, Morris R. Tarzan: A peer-to-peer anonymizing network layer. In *Proc. the 9th ACM Conference on Computer and Communications Security* (*CCS*), Washington DC, USA, 2002, pp.193–206.

[32] Liu D, Chi C-H, Li M. Normalizing traffic pattern with anonymity for mission critical applications. In *Proc. the 37th Annual Simulation Symposium,* Arlington, USA, 2004, pp.293–299.

[33] Berthold O, Langos H. Dummy traffic against long term intersection attacks. In *Proc. Privacy Enhancing Technologies Workshop* (*PET*), San Francisco, CA, USA, 2002, pp.199–203.

[34] Dingledine R, Freedman M J, Molnar D. The free haven project: Distributed anonymous storage service. In *Proc. Workshop on Design Issues in Anonymity and Unobservability,* Berkeley, California, USA, 2000, pp.67–95.

[35] Han J, Liu Y, Xiao L, Xiao R, Ni L M. A mutual anonymous peer-to-peer protocol design. In *Proc. the 19th International Parallel & Distributed Processing Symposium* (*IEEE IPDPS*), Denver, CA, USA, 2005, p.68.1.

[36] Han J, Liu Y. Mutual anonymity for mobile peer-to-peer systems. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*). (To appear)

[37] Han J, Liu Y. Rumor riding: Anonymizing unstructured peer-to-peer systems. In *Proc. IEEE International Conference on Network Protocols* (*ICNP*), Santa Barbara, California, 2006, pp.22–31.

[38] Serjantov A. Anonymizing censorship resistant systems. In *Proc. the 1st International Workshop on Peer-to-Peer Systems,* Cambridge, MA, USA, 2002, pp.111–120.

[39] Waldman M, Rubin A D, Cranor L F. Publius: A robust, tamper-evident, censorship-resistant web publishing system.

In *Proc. the 9th USENIX Security Symposium,* Denver, Colorado, USA, 2000, pp.59–72.

[40] Roger Dingledine. The free haven project: Design and deployment of an anonymous secure data haven [Thesis]. MIT, June 2000.

[41] Sherwood R, Bhattacharjee B, Srinivasan A. P^5: A protocol for scalable anonymous communication. In *Proc. IEEE Symposium on Security and Privacy,* Oakland, California, USA, 2002, pp.58–70.

[42] Levine B N, Shields C. Hordes: A multicast based protocol for anonymity. *Journal of Computer Security*, 2002, 10(3): 213–240.

[43] Wang Y, Dasgupta P. Anonymous communications on the Internet. In *Proc. the IASTED International Conference on Communication, Network and Information Security,* Phoenix, Arizona, USA, 2005, p.499.

[44] Waters B R, Felten E W, Sahai A. Receiver anonymity via incomparable public keys. In *Proc. the 10th ACM Conference on Computer and Communications Security* (*ACM CCS*), Washington DC, USA, 2003, pp.112–121.

[45] Luo X, Qin Z, Han J, Chen H. DHT-assisted probabilistic exhaustive search in unstructured P2P networks. In *Proc. the 22nd IEEE International Parallel and Distributed Processing Symposium* (*IEEE IPDPS*), Miami, Florida, USA, 2008. (To appear)

[46] Nandan A, Pau G, Salomoni P. GhostShare — Reliable and anonymous P2P video distribution. In *Proc. the 1st IEEE Global Telecommunications Conference* (*GlobeCom*) *Workshops,* Dallas, Texas, USA, 2004, pp.200–210.

[47] Clarke I, Sandberg O, Wiley B, Hong T W. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. Workshop on Design Issues in Anonymity and Unobservability,* Berkeley, CA, USA, 2000, pp.44–66.

[48] Wright M K, Adler M, Levine B N, Shields C. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security* (*TISSEC*), 2004, 7(4): 489–522.

[49] Shmatikov V. Probabilistic analysis of anonymity. In *Proc. the 15th IEEE Computer Security Foundations Workshop* (*CSFW*), Cape Breton, Nova Scotia, Canada, 2002, pp.119–128.

[50] Zhu Y, Bettati R. Anonymity vs. information leakage in anonymity systems. In *Proc. the 25th IEEE International Conference on Distributed Computing Systems* (*ICDCS*), Columbus, Ohio, USA, 2005, pp.514–524.

[51] Guan Y, Fu X, Bettati R, Zhao W. A quantitative analysis of anonymous communications. *IEEE Transaction on Reliability*, 2004, 53(1): 103–115.

[52] Levine B N, Reiter M K, Wang C, Wright M. Timing attacks in low-latency mix systems. In *Proc. the 8th International Conference on Financial Cryptography,* Key West, Florida, USA, 2004, pp.251–265.

[53] Serjantov A, Sewell P. Passive attack analysis for connection-based anonymity systems. In *Proc. European Symposium on Research in Computer Security* (*ESORICS*), Norway, 2003, pp.116–131.

[54] Fu X, Graham B, Xuan D, Bettati R, Zhao W. Analytical and empirical analysis of countermeasures to traffic analysis attacks. In *Proc. IEEE International Conference on Parallel Processing* (*ICPP*), Kaohsiung, 2003, pp.483–492.

[55] Fu X, Zhu Y, Graham B, Bettati R, Zhao W. On flow marking attacks in wireless anonymous communication networks. In *Proc. the 25th International Conference on Distributed Computing Systems* (*IEEE ICDCS*), Columbus, Ohio, USA, 2005, pp.493–503.

[56] Guan Y, Li C, Xuan D, Bettati R, Zhao W. Preventing traffic analysis for real-time communication networks. In *Proc. IEEE Military Communications* (*MILCOM*), Atlantic City, NJ, USA, 1999, vol.1, pp.744–750.

[57] Murdoch S J. Danezis G. Low-cost traffic analysis of Tor. In *Proc. IEEE Symposium on Security and Privacy,* Oakland, California, USA, 2005, pp.183–195.

[58] Breslau L, Cao P, Fan L, Phillips G, Shenker S. Web caching and Zipf-like distributions: Evidence and implications. In *Proc. IEEE INFOCOM,* New York, USA, Vol.1, 1999, pp.126–134.

[59] Lu L, Han J, Liu Y, Hu L, Huai J, Ni L M, Ma J. Pseudo trust: Zero-knowledge authentication in anonymous P2Ps. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*). (To appear)

[60] Han J, Liu Y. Dubious feedback: Fair or not? In *Proc. International Workshop on Peer-to-Peer Information Management* (*P2PIM*), Hong Kong, 2006, Article No.49.

[61] Tan G, Jarvis S. A payment-based incentive and service differentiation scheme for peer-to-peer streaming broadcast. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*), 2007, 19(7): 940–953.

[62] Xiao L, Zhu Y, Xu Z, Ni L. Incentive-based decentralized scheduling for computational grids. *IEEE Transactions on Parallel and Distributed Systems* (*TPDS*). (To appear)

Ren-Yi Xiao received his Master's degree from Beihang University in 1990. He has been working at NSF of China as a project manager since 1993. He was a visiting scholar at UT Austin, USA, from March 2000 to September 2000. His research interests include self-organizing network, peer-to-peer computing, sensor networks, pervasive computing, and network security.