Secure Rumour Riding Protocol For Decentralised Peer To Peer Systems

Rekha.G [1], Dr V. Saravanan[2]
Department of Information Technology
[1] PG Student, Hindusthan College of Arts and Science, Coimbatore, India
[2] HOD & Professor, Hindusthan College of Arts and Science, Coimbatore, India

Abstract- *Although anonymizing Peer-to-Peer (P2P) systems often incurs extra costs in terms of transfer efficiency, many systems try to mask the identities of their users for privacy considerations. Existing anonymity approaches are mainly path-based: peers have to pre-construct an anonymous path before transmission. The overhead of maintaining and updating such paths is significantly high. In this project, we propose Rumour Riding (RR), a lightweight mutual anonymity protocol for decentralized P2P systems. RR employs a random walk scheme which frees initiating peers from the heavy load of path construction.*

RR also takes advantage of lower cryptographic overhead by mainly utilizing a symmetric cryptographic algorithm to achieve anonymity. We demonstrate the effectiveness of this design through trace-driven simulations. The analytical and experimental results show that RR is more efficient than existing protocols.
In proposed system we applying the Rumour Riding(RR) concept to overcome the above-mentioned problems. It's a lightweight and non-path-based mutual anonymity protocol for decentralized P2P systems. It can't attack by attackers very easily. Symmetric cryptographic algorithm is used in Rumour riding.

Keywords- P2P , Anonimity , Centralised , etc.

I. INTRODUCTION

Existing anonymity approaches are mainly path-based: peers have to pre-construct an anonymous path before transmission. The overhead of maintaining and updating such paths is significantly high.

In this project, we propose Rumour Riding (RR), a lightweight mutual anonymity protocol for decentralized P2P systems. It employs a random walk scheme which frees initiating peers from the heavy load of path construction.

RR also takes advantage of lower cryptographic overhead by mainly utilizing a symmetric cryptographic algorithm to achieve anonymity.

II. SYSTEM ANALYSIS

EXISTING SYSTEM

Peer-to-peer (P2P) networks, such as Napster, Gnutella, and Bit Torrent, have become essential media for information dissemination and sharing over the Internet. Recently, a number of P2P users have encountered problems caused by being traced on non-anonymous P2P systems due to their plain-text query messages and direct-downloading behaviors. Hence, the requirement for anonymity has become increasingly essential in current P2P applications for both content requesters and providers. Most, if not all of them, deliver messages via non-traceable paths comprised of several anonymous proxies or middle agent peers. In these approaches, known as path-based approaches, users usually need to construct anonymous paths before transmissions. All nodes in the path cooperate to forward data to a receiver. Data is pre-wrapped by the initiator in a layered-encryption packet (usually using asymmetric cryptographic algorithms, such as RSA), which will be peeled off along the path to the receiver. Although path-based protocols provide strong anonymity, they have the following problems:

DISADVANTAGES

1. Pre-construction of paths requires users to obtain a large number of IP addresses and public keys from other peers in advance. Both the collection of information and the preparation of packets incur high costs.

2. Initiators have to periodically update middle nodes along the anonymous paths. An invariable path might otherwise become increasingly vulnerable under the analysis of attackers. In addition, users often expect to extend the length of anonymous paths, as a longer path entails a higher degree of anonymity. Both of these requirements increase the maintenance and update overhead.

3. In highly dynamic P2P systems, peers randomly join and leave. If a chosen node goes offline, the whole path fails, and such a failure is often undetected by the initiator.

PROPOSED SYSTEM

We propose a non-path-based anonymous P2P protocol called RumourRiding (RR). The design goal is twofold: first, to eliminate the huge overhead of path construction and maintenance; second, to use a symmetric cryptographic algorithm to replace the asymmetric one so as to reduce the cryptographic overhead and make the protocol more practical. we first let an initiator encrypt the query message with a symmetric key, and then send the key and the cipher text to different neighbors. The key and the cipher text take random walks separately in the system, where each walk is called a Rumour. Once a key Rumour and a cipher Rumour meet at some peer, the peer is able to recover the original query message and act as an agent to issue the query for the initiator. A similar idea is also employed during the query response, confirm, and file delivery processes.

ADVANTAGES

1. It improves the efficiency and effectiveness of the design.
2. This provide better result when compare with previous process.
3. It used random walk mechanism to send data to receiver.

III. MODULE DESCRIPTION

INITIATOR

This is the sender module that will make the process start, some of the initiator process are make the query request with the key to decrypt it, Message with its decrypt key.

RESPONDER

Responder is the receiver, can retrieve the data by make the key first, the responder should response to the query request then can get the key with its origial message.

QUERY REQUEST AND RESPONSES

This module helps to make the connectivity between the sender and receiver at the initial stage, query request will be send by the initiator to make the connection with desired destination. If the responder response query message then there will be a communication will be happened.

MESSAGE AND ITS KEY

It is the original data which the initiator willing to send. This message will transfer in the network path and the key need to decrypt is travel in another path, when both this will meet the node can get the data.

ADMIN / REPORTS.

This module is the overall controller of the network, this makes all users authenticate and maintain the Rumour path with the reports in detail.

IV. SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

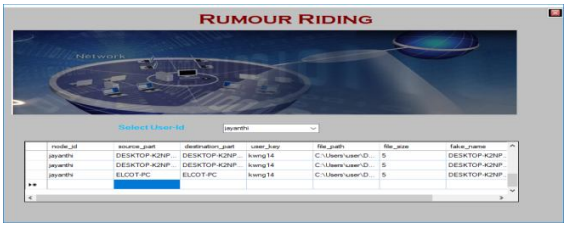

Figure 1. Data Transaction

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

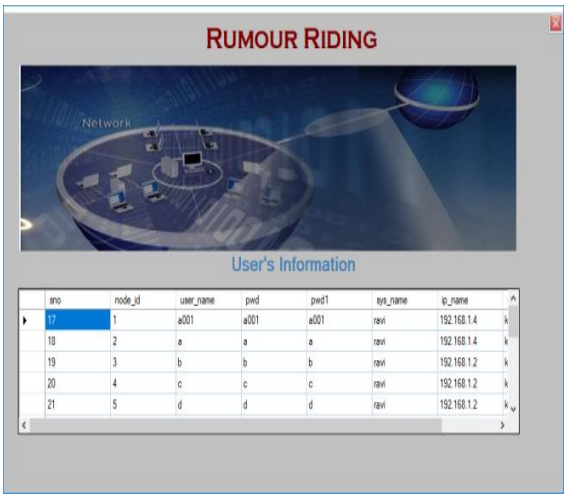

Figure 2. Display of User Information

The project is implemented by accessing simultaneously from more than one system and more than one window in one system. The application is implemented in the asp.net under the Windows 7 Professional and accessed from various clients.

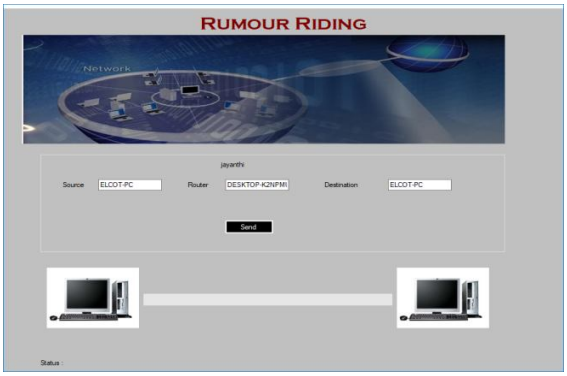

Figure 3. Identifying Fake Path

V. CONCLUSION & FUTURE ENHANCEMENT

We propose a lightweight and non-path-based mutual anonymity protocol for P2P systems, Rumour Riding (RR). Employing a random walk concept, RR issues key Rumoursand cipher Rumours separately, and expects that they meet in some random peers.

The results of trace-driven simulations and simple implementations show that RR provides a highdegree of anonymity and outperforms existing approaches in terms of reducing the traffic overhead and processing latency. We also discuss how RR can effectively defend against various attacks. Future and ongoing work includes accelerating the query speed, introducing mimic traffic to confuse attackers, and optimizing the k and L combination to further reduce the traffic overhead.

We will also investigate other security properties of RR, such as the unlink ability, information leakage, and failure tolerance when facing different attacks. It would also be interesting to explore the possibility of implementing this lightweight protocol in other distributed systems, such as grid systems and ad-hoc networks.

REFERENCES

[1] D. Chaum, "Untraceable electronic mail return addresses and digital pseudonyms", Communications of the ACM, 1981.
[2] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, "Web caching and Zipf-like distributions: evidence and implications", Proceedings of IEEE INFOCOM, 1999.
[3] D. Goldschlag, M. Reed, P. Syverson, "Onion routing", Communications of the ACM, 1999.
[4] M. K. Reiter, A. D. Rubin, "Crowds: anonymity for web transactions", ACM Transactions on Information and System Security, 1998.
[5] S. Saroiu, P. Gummadi, S. Gribble, "A measurement study of Peer-to-Peer file sharing systems", Proceedings of Multimedia Computing and Networking, 2002.
[6] K. Sripanidkulchai, "The popularity of Gnutella queries and its implications on scalability", Proceedings of The O'Reilly Peer-to-Peer and Web Services Conference, 2001.
[7] V. Scarlata, B. N. Levine, C. Shields, "Responder anonymity and anonymous Peer-to-Peer file sharing", Proceedings of IEEE ICNP, 2001.
[8] I. Abraham, D. Malkhi, "Probabilistic quorums for dynamic systems", Proceedings of DISC, 2003.
[9] R. Sherwood, B. Bhattacharjee, A. Srinivasan, " P 5 : A protocol for scalable anonymous communication ", Proceedings of IEEE Symposium on Security and Privacy, 2002.
[10] S. Sen, J. Wang, "Analyzing Peer-to-Peer traffic across large networks", Proceedings of ACM SIGCOMM Internet Measurement Workshop, 2002.
[11] L. Xiao, Z. Xu, X. Zhang, "Low-cost and reliable mutual anonymity protocols in Peer-to-Peer networks", IEEE Transactions on Parallel and Distributed Systems, 2003
[12] R. Dingledine, N. Mathewson, P. Syverson, "Tor: the second-generation onion router", Proceedings of the USENIX Security Symposium, 2004.

WEBSITES:

[13] http:// www-2.cs.cmu.edu/~Kunwadee/research/p2p/gnutella.html.
[14] http://www.cse.ust.hk/~jasonhan/RR-TR.pdf.
[15] http:// www.propagandactritic.com/articles/about.html.
[16] http://www.sei.cmu.edu./domain-engineering/usecasediagram.html
[17] http://en.wikipedia.org/wiki/windows-XP
[18] http://www.sys-design.com