K. Sudhakar, G.S. Raj, P.Visu, M. Saleem Babu, Koteeswaran. S / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
www.ijera.com Vol. 2, Issue 1, Jan-Feb 2012, pp.754-757

Secure File Transfer over Peer -To-Peer Network

K. Sudhakar[1], G.S. Raj[2], P.Visu[3], M. Saleem Babu[4], and Koteeswaran. S[5]

[1]PG Scholar, [2]Asst.Professor, [3,5]Research Scholar, [4]Professor
Dept of CSE, Vel Tech Dr. RR & Dr. SR Technical University, Chennai.

ABSTRACT

One of the threats which cause defects to the data is Byzantine attack. The minimum probability of attack may cause maximum probability of defect. Omission and commission are the two effects of Byzantine attack. Omission causes the loss of packets while transferring the packets. Commission will lead to the collapse of the data sent. Entire file may change or the content of the file is changed. To prevent these attacks we are going to propose a model which prevents the transfer of data from the threats. In this model the file is detected at each router. Once the router detects the infected packets, it will be discarded at that router itself.

Keywords— *Byzantine, Web service, Business entity and Business service*

1. INTRODUCTION

When transferring the files in a peer to peer system may pave the way to many attacks. Here we are going to counteract Byzantine attacks. Two main causes of the Byzantine attacks are omission and commission. In omission failures may occur during the transmission that is sending and receiving the files and in commission it may lead to incorrect or irrespective response. To overcome this we are going to implement the algorithm called Diffie Hellman Algorithm to provide signature to the data sent. It can provide secured file transfer in a peer to peer network.

In this algorithm we are going to take three values namely p, g, (a or b), Where p is the assumed prime value and then the g is the generative value assumed by the administrator. And then 'a' which is the secret value assumed at the client side during runtime. The value for 'b' which is said to be as the secret value assumed at the server side. After that that the A and B values are calculated by using the formula.

$$A=ga \% p \quad (1)$$

$$B=gb \% p \quad (2)$$

After the calculation of equation 1 value and then equation 2 value, the answer is passed to the server and to client vice versa. After receiving those values the final result will be calculated at the both end for the purpose of key verification. Here it involves,

$$Kclient=Ba \% p \quad (3)$$

$$Kserver=Ab \% p \quad (4)$$

Finally it is found that equation 3 and 4 are equal. If it is not equal then the key generated is illegal and the authentication is denied. If they are equal the server proceeds the following action. The file is transmitted without the omission and commission attacks caused by Byzantine.

2. RELATED WORK

Network coding [1], an alternative to the traditional forwarding paradigm, allows algebraic mixing of packets in a network. It maximizes throughput for multicast transmissions [2], [3], [4], robustness against failures [5] and erasures [6]. Random linear network coding (RLNC), in which nodes independently take random linear combination of the packets, is sufficient for multicast networks [7], and is suitable for dynamic/unstable networks, such as peer-to-peer (P2P) networks [8], [9].A P2P network is a cooperative network in which storage and bandwidth resources are shared in a distributed architecture. This is a cost-effective and scalable way to distribute content to a large number of receivers. One such architecture is the Bit Torrent system [10], which splits large files into small blocks. After a node downloads a block, it acts as a source for that particular block. The main challenges in these systems are the scheduling and management of rare blocks. Despite their desirable properties, network coded P2P systems are particularly susceptible to Byzantine attacks [11],[12], [13] – the injection of corrupted packets into the information flow[14].

3. OVERALL SYSTEM ARCHITECTURE

Fig 3.1 describes the overall system architecture. It contains

3.1 User Interface Layer (UI)

User Interface Layer creates an interface between the user and the Application. The user interface layer contains web forms, Master Page, style sheets etc.. User can navigate through these forms and they can communicate to and fro with the Application through this user interface layer.

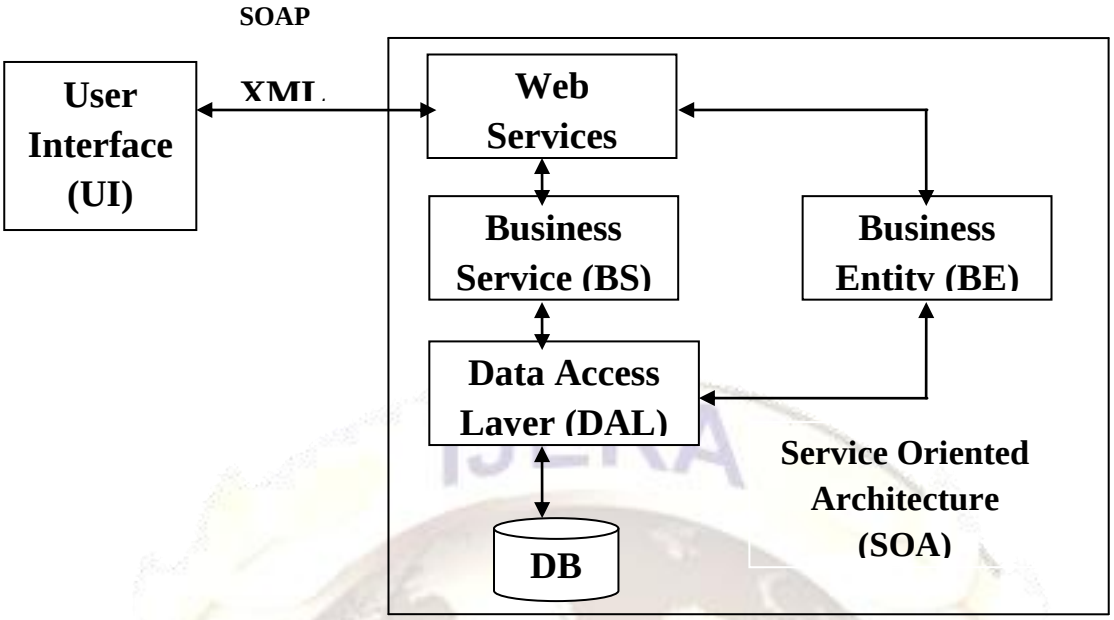

Fig 3.1 : Overall system Architecture

3.2 Data Communication

The User Interface (UI) will communicate with the Business Service (BS) Layer and the Business Entities (BE) Layer. From the UI the value will be set to the BE Layer and then the value will be passed to the BS Layer. The BS Layer passes the values from the UI to the DAL. The values passed may be object of BE Layer or any other values like string, int, etc.

3.3 Business Entity (BE)

The data entered by the user through the web form in the user interface layer are known as entities and they should be encapsulated for security purpose. For this purpose we are using this Business Entity Layer. Here all the variables for entities are declared as private to prevent access to unauthorized members. The properties of the entities are set by using the get and the set method.

3.4 Business Service

Business Service (BS) Layer act as a bridge between the UI and the DAL. This Layer dose not perform any logical operation, it's just a service layer

which services UI and DAL. A business service is a function of the business that is offered to one or more clients. Those clients are often internal, because this often applied to supporting functions.

3.5 Data Access Layer (DAL)

DAL Communicates with BS Layer and BE Layer. The layer minimizes the ADO.NET work by SQL Helper Class. The SQL Helper handles all the Database related activities and will reduce the development time and ensure the coding standard. Data Access Layer , which helps separate data-access logic from your business objects.

4. SYSTEM WORKING

Fig. 4.1: describe system working architecture. It consist of

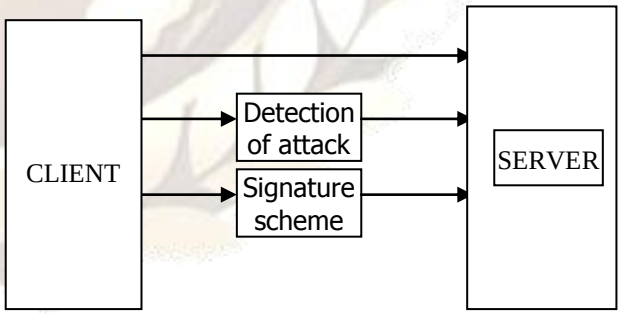

Fig. 4.1: System Architecture

4.1 Login

The first step is to provide authentication to the user. If the user is authenticated he will enroll his username and password and he starts the proceedings.If the user is new he will register his details and create a new account. Then he is an authenticated user and he can access the process.

4.2 Peer to Peer File Transfer

This is the main process of the project. A file is selected from a peer and it is transferred to another peer. The

systems have to be connected in a LAN or any sort network connection. Network rules have to be followed to send the file from one system to another.

4.3 Detection of Byzantine Attack

The file is transferred. After it is received it has to be detected whether there is any attack. If the packets are lost or any drastic change in the data it has to be confirmed that file is attacked by any threat. To find the attack of Byzantine, peer to peer detection is carried out.

4.4 Impact of the Byzantine Attack

For the small probability of attack there will be a large probability of defect in the data. The various impacts of Byzantine attack is overviewed. The two major impacts of Byzantine are,

- Omission
- Commission

4.4.1 Omission

The impact of omission is there will be a loss of packet at the receiver side. The transferred packets may be defected and it may lose.

4.4.2 Commission

Commission causes in the modification of the data. It will be received in totally different format.

4.5 Signature Scheme

To prevent the data from the threat it has to be provided with the signature scheme. The signature scheme we use here is Diffie-Hellman algorithm. Using this signature scheme we are going to protect the data from the threat.

5. CONCLUSION AND RESULTS

In this paper, we studied the problem of Byzantine attacks in network coded P2P networks. We used randomly evolving graphs to characterize the impact of Byzantine attackers on the receiver's ability to recover a file. As shown by our analysis, even a small number of attackers can contaminate most of the flow to the receivers. Motivated by this result, we proposed a novel signature scheme for any network using RLNC. The scheme makes use of the linearity of the code, and it can be used to easily check the validity of all received packets. using this scheme, we can prevent the intermediate nodes from spreading the contamination by allowing nodes to detect contaminated data, drop them, and therefore, only transmit valid data. The Byzantine probability has been shown in fig 5.1

The Byzantine probability P_b describes possible Byzantine probability ratio in static tracker list. The Blocking probability specifies the ratio of detecting and blocking the Byzantine attack. We emphasize that there is no need of retransmission for the dropped data since the receivers can perform erasure correction, which is computationally cheaper than error correction. We analyzed the cost and benefit of the signature scheme, and compared it with various detection schemes. We showed that the overhead of our scheme is low. Furthermore, when the probability of attack is high, it is the most bandwidth efficient. However, if the probability of attack is low, generation-based detection schemes are more appropriate.

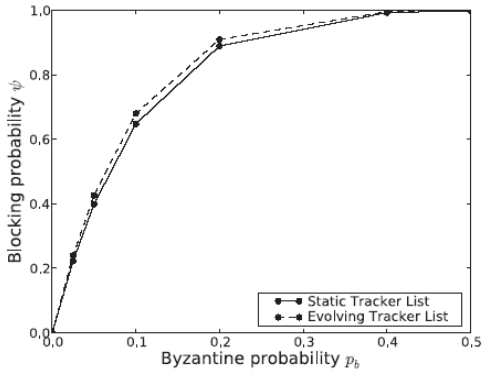

Fig. 5.1. Byzantine Probability Vs Blocking Probability

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, 2000.

[2] T. Ho, M. M´edard, M. Effros, and D. Karger, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE ISIT*, Kanagawa, Japan, July 2003.

[3] Z. Li and B. Li, "Network coding: the case of multiple unicast sessions," in *Proc. 42nd Annual Allerton Conference on Communication Control and Computing*, September 2004.

[4] D. Lun, M. M´edard, and R. Koetter, "Network coding for efficient wireless unicast," in *Proc. International Zurich Seminar on Communications*, Zurich, Switzerland, February 2006.

[5] R. Koetter and M. M´edard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, pp. 782–795, 2003.

[6] D. Lun, M. Medard, R. Koetter, and M. Effros, "*On coding for reliable communication over packet networks,"* Physical Communication, vol. 1, no. 1, pp. 3–20, 2008.

[7] T. Ho, M. M´edard, R. Koetter, M. Effros, J. Shi, and D. R. Karger, "A random linear coding

approach to mutlicast," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4413–4430, 2006.

[8] S. Aceda´nski, S. Deb, M. M´edard, and R. Koetter, "How good is random linear coding based distributed network storage?" in *Proc. 1st Netcod,Riva del Garda*, Italy, April 2005.

[9] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE INFOCOM*, Miami, FL, March 2005.

[10] "Bittorrent file sharing protocol," http://www.BitTorrent.com.

[11] R. Perlman, "Network layer protocols with Byzantine robustness," Ph.D.dissertation, Massachusetts Institute of Technology, Cambridge, MA, October 1988.

[12] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Symposium on Operating Systems Design and Implementation (OSDI)*, February 1999.

[13] L. Lamport, R. Shostak, and M. Pease, "*The Byzantine generals problem*," ACM Trans. Programming Languages and Systems, vol. 4, pp. 382–401, 1982.

[14] MinJi Kim, Lu´ısa Lima, Fang Zhao, Jo˜ao Barros, Muriel M´edard, Ralf Koetter, Ton Kalker, Keesook J. Han "On Counteracting Byzantine Attacks in Network Coded Peer-to-Peer Networks " in *Proc. IEEE IJASC*, 2010.