International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

LEPMA: Lightweight Extensible Authentication Protocol based on Mobile Agent

Yojana
Umesh Kumar

Abstract

Mobile agent technology is becoming more popular and has been implemented in many areas. This paper proposes a new LEAP mobile agent based model and their corresponding algorithm. LEAP mobile agent based model(LEPMA), algorithms discussed in this paper. On the internet, as the traffic and remote interaction time increases at a screaming speed, proposed a technique to decrease the speed of the same factors. In terms of network traffic and remote interaction time comparison analysis of proposed algorithm and comparison with the traditional client server based mechanism[4]. The proposed algorithm reduces the traffic and remote interaction time to a great extent as compared to the client server based mechanism. The paper proposes a new Mobile agent based model, algorithms and generate equations corresponding to algorithms.

Keywords:

Client Server;
Mobile Agent;
LEAP;
LEPMA.

Author correspondence:

Yojana[1],Umesh Kumar[2], Sapna Gambhir[2]
Department of Computer Engineering ,YMCA University Of Science and Technology
Faridabad, India
Email: yojana.011@gmail.com[1],umesh554@gmail.com[2]

1. Introduction

User authentication is a service crucial for many electronic transactions. Without a secure verification of users, it would be impossible to provide many services both on the Internet and during everyday life. For the verification of identities and person's authorizations need authentication methods. Authentication is the technique which allows a sender and a recipient to approve one another. It can be done by providing a username and a password to identify themselves against a legitimate record in the database to check the combination is correct.A communication protocol that is used to transfer authentication data between two entities is known as authentication protocol. It is up to the authentication procedures defined to protect the server's assets from getting unauthorized access and it should not be costlier than the information to be secured [1]. The establishment of the identities of the participating entities and to distribute secret session keys are the major goals of an authentication protocol.Various Authentication Protocols are: EAP, PAP, CHAP, MS-CHAP.The type of EAP protocol depends upon the type of security required and the level of security required.

EAP is an authentication framework that is designed to run on the data link layer where IP connectivity is not available. It provides a basic request/response protocol framework over which various EAP methods can be implemented.Some authentication methods are predefined like LEAP, TLS, POTP, MD5, PSK, TTLS and SIM.Initially EAP wasinvented to work connections of the nature of point to point.The same EAP later on was adapted by IEEE 802 wired networks as well as wireless LAN networks.LEAP uses a modified version of MS-CHAP, an authentication protocol in which consumer credentials are now not strongly protected. More grounded confirmation conventions contract a salt to make more grounded the qualifications contrary to listening in all through the verification procedure [3]. This protocol works on the client server architecture. The problem with this is

that client needs to be connected to the server continuously without any network interruption. The bandwidth requirement is higher in client server based architecture a compared to the mobile agent based architecture.

This paper is organized as follows. In the following section discuss various problems present in client server architecture and EAP-LEAP protocol. In the next section, we explain a proposed work in which LEPMA model and its corresponding algorithm is mentioned after that numerical analysis done for comparison of client server architecture and mobile agent based model on the basis of two parameters traffic analysis and remote interaction time and in last section we end with some conclusions.

2. Problem Statement:

There are number of problems associated with the client server architecture as compared with the mobile agent based architecture. As EAP is also based on client server based architecture, these problems are also associated with the EAP also. Some of these problems are:

- ***Higher bandwidth requirement***
- ***Number of message exchanges are higher***
- ***Traffic generated is more***
- ***Dynamic change in code or procedure***

3. Proposed Work:

In the next section the LEAP based on mobile agent model is proposed. To overcome allproblems present in client server architecture proposed a mobile agent framework LEPMA.

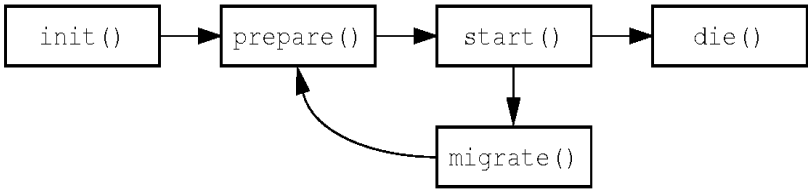
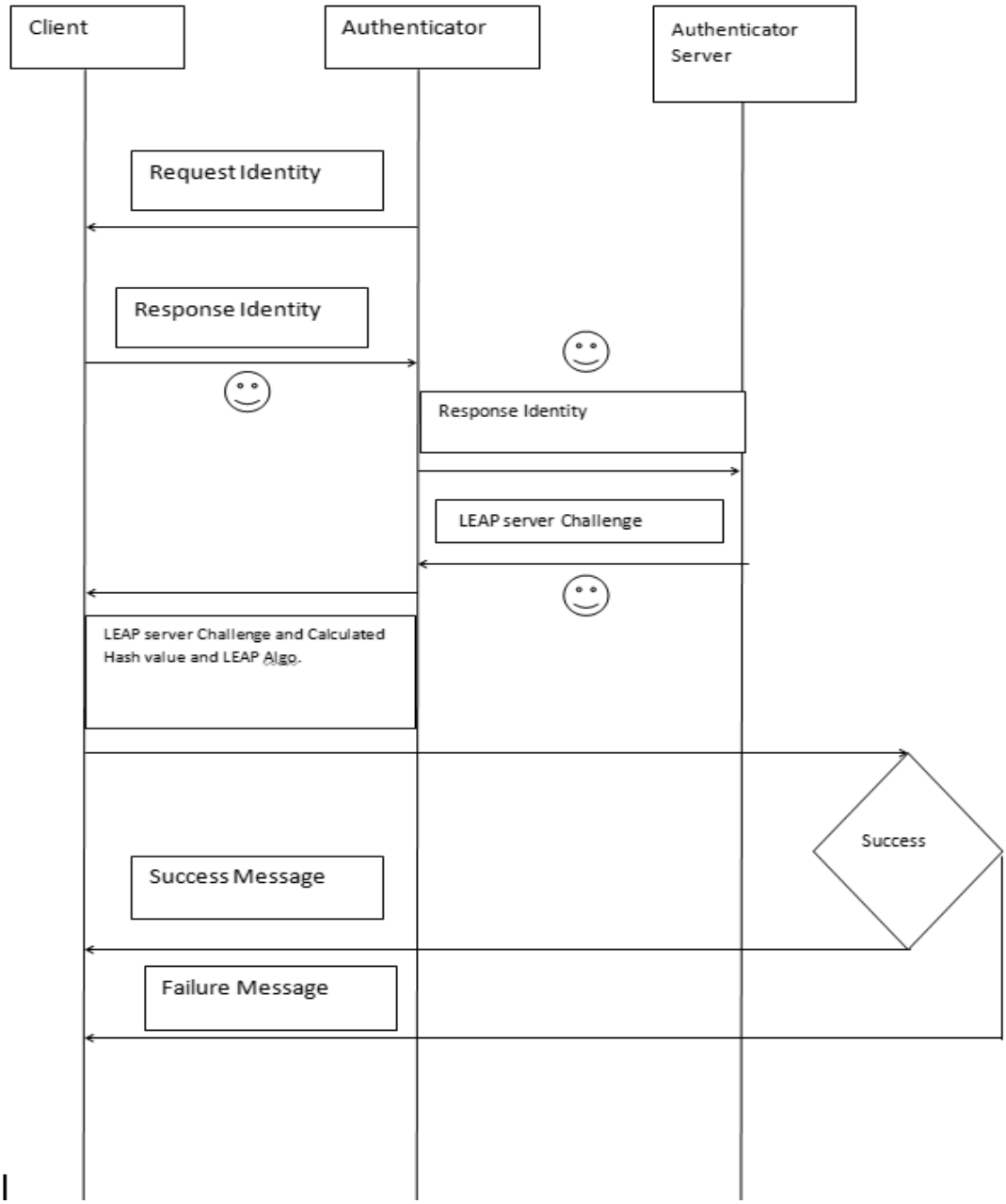


Fig 1. Life Cycle of Mobile Agent

In our previous research mobile agent based framework for wireless authentication (MABFWA) [17],which was designed to provide a framework to support multiple authentication protocols was proposed. Framework was designed based on mobile agent and with the assumption that participating entities has agent technology like AGLET environment installed.

This section proposes LEAP on the (MABFWA) that we call it LEPMA (LEAP-EAP Protocol using Mobile Agent) to overcome disrupt quality of client/server model is that of scaling. Figure 2 shows the server side authentication using the mobile agent. As soon as the client comes into the range of the authenticator, it requests the identity and client responds with the identity. Authentication server then passes the challenge to the client through authenticator. Client responds with the challenge result in a hash value to the authentication server. Server responds with the success or failure message to the client, depending upon the hash value.

International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A



/ Fig 2 LEAP Server side Authentication using MA

International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

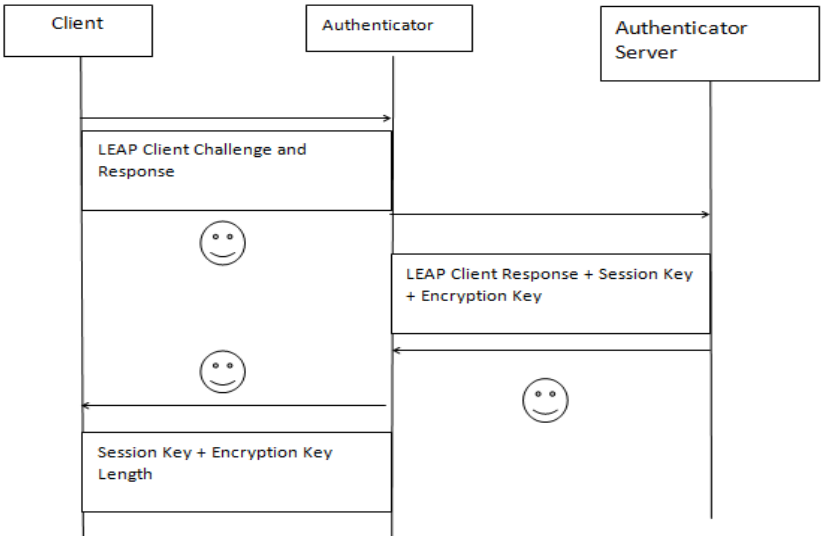


Fig 3 LEAP Client side Authentication using MA.

In the proposed model, first of all the device needs to be shared identity or share the successful shared identity between device.

Algorithm I: Client

Input: 1. Client Address.
2. Authenticator Address.
3. LEAP Client Challenge.

Output: 1. Success or Failure of Authentication.
2. Calculated value for LEAP Client Challenge.

1.bit = 0;
1.1fori = 1,2,3…. M rounds do
1.1.1 Handle EAP-Request/Identity message and provide EAP-response/Identity message;
1.2 if (success == TRUE) then Set bit = 1;
1.2.1 break;
1.3else
1.3.1 Provide correct Identity;
1.4 end if
1.5end for
2.while (bit != 0) do
2.1for j = 1,2,…,M rounds do
2.1.1 Handle LEAP Challenge Request and Calculate the Hash value string with Leap Algo;
. **2.1.2** Send MA to authenticator having "Calculated Hah Value and LEAP " to the authenticator
2.2 break;
2.3 end for
3. Receive EAP-Success/Failure message from authenticator server;
3.1 if (Hash Value at Client side == Hash Value at Authenticator Server) then
3.1.1receive Success
3.2else
3.2.2receive Failure
3.3 end if
3.4 end while

After successful connection by sharing a identity key then they are able to communicate with each other. Identity request shared between authenticator and client then successfully identity shared between them.

International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Handle LEAP Challenge Request and Calculate the Hash value string with Leap Algo and Mobile Agent (MA) send to authenticator having calculated Hash Value and LEAP Algo. From authentication server it receive EAP-Success/Failure message client respond according to the request.

Algorithm II: Authenticator

Input: 1. Client Address.
2. Authenticator Address.
3. Authenticator server Address
4. LEAP Server Challenge
5. LEAP Client Challenge

Output: 1. Success or Failure of Authentication.
2. LEAP Client Response.
3. LEAP Server Response.

1.while (N>0) do
1.1fori = 1,2, …, M rounds do
1.1.1 Send EAP-Request/Identity message and Handle EAP-Response/Identity from client;
1.2if (EAP-Response/Identity == found) then
1.2.1 Send EAP-Response to the authenticator server;
1.3else
1.4goto step 3;
1.5 end if
1.6end for
2.for j = 1,2, …., M rounds do
2.1 while (EAP-Request != null) do
2.1.2.Receive MA containing challenge string from Authenticator server;
2.1.3 Send MA containing EAP-Request having challenge to client;
2.1.4 Receive EAP-Response from the client;
2.1.5 Send EAP-Response to the server;
2.2 end while
2.3 end for

Algorithm III: Authentication Server

Input: 1. Authenticator Address.
2. Authenticator Server Address.
3. RADIUS message for verification.
4. LEAP Client Challenge.

Output: 1. Success or Failure of Authentication.
2.Calculated value for LEAP Client Response
.
1.while (N>0) do
1.1fori = 1,2,3,...,M rounds do
1.1.1send EAP-Request MA to the client via authenticator;
1.1.2 Receive EAP-Success/Failure message from authenticator;
1.2 if (hash value at client side= hash value at server side) then
1.2.1 Send EAP-Success message to client;
1.3else
1.3.1 Send EAP-Failure message to client;
1.4 end if
1.5 end while
2. Receive MA containing LEAP client Challenge and Response from authenticator
3.Send MA to Client via authenticator having Session Key and Encryption Key.

In algorithm for authenticator server, Authenticator send EAP Request MA to the client via authenticator. If hash value at client side is equal to the hash value at server side then send EAP success message to client otherwise failure message. Receive MA containing LEAP client Challenge and response from authenticator. In the last it sendMA to client via authenticator having session key and encryption key.

International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

4. Result and Analysis

Numerical Analysis of Client Server and Proposed Mobile Agent Technique

Proposed technique is compared with the existing client server approach against parameters like cost of management and remote interaction time. Numerical analysis of both the approaches has been done.

- **Client Server approach**

In Client-Server approach the Authenticator receives the traffic from multiple clients seeking authentication within the network. Following equation is used to calculate the complete traffic around authenticator within the network:

$$T_{fc}\ C_{cl-sr}^{mmt} = \sum_{i=1}^{n} \begin{Bmatrix} (Srq + Srs) * a * \\ Average\ session\ no. + (Srq + Srs) * b \end{Bmatrix} \quad ...(i)$$

where,

$T_{fc}\ C_{cl-sr}^{mmt}$ = Management Cost in network traffic.

Srq = Client to Server Request size and

Srs = Server to client size

Avg. = Each client's avg. session no.

a= Depending upon protocol,number of message exchanges between client and authenticator.

n= Clients number

b= Depending upon protocol, number of calls to authentication server .

Remote interaction time enforcedby the authenticator to approve the clients over the network will depend upon the bandwidth available and will be calculated as:

$$Remote\ interaction\ time = \frac{Traffic}{Bandwidth} \quad ...(ii)$$

$$T^m C_{cl-sr}^{rm} = \sum_{j=1}^{n} \frac{(Srq + Srs)}{BW_j} + 2Lt_j \quad ...(iii)$$

where,

$T^m C_{cl-sr}^{rm}$ = In client server architecture, Remote interaction time for one message exchange with n number of clients.

$2Lt_j$ = Latency time between authenticator and j[th] client.

- **Proposed MA based approach**

In proposed mobile agent based approach the cost of management in terms of network traffic generated at the authenticator will be calculated as follows:

$$T_{fc} C_{ma}^{mmt} = \left\{ S_{ma} + \sum_{j=1}^{n} S_{pr} \right\} \quad ...(iv)$$

where,

$T_{fc}C_{ma}^{mmt}$ =For Mobile agent architecture the management cost in terms of network traffic.

Sma= Mobile agent size carrying the authentication algorithm code to be executed.

Spr = Partial result generated at each client.

Here we can have single user authentication and multiuser authentication also. For single user authentication the traffic will be

$$T^m C_{ma}^{rm} = \{S_{ma} + S_{pr}\} \qquad \ldots(v)$$

So, depend upon the size of the mobile agent from above equation amount of traffic generated at the authenticator.

Remote interaction time required by the authenticator to validate the clients over the network will depend upon the bandwidth available and will be calculated as follows for MA:

$$T^m C_{ma}^{rm} = \sum_{j=1}^{n} \frac{(S_{ma} + S_{pr})}{Bw(j-1,j)} + Lt(j-1,j) \qquad \ldots(vi)$$

where,
$Lt(j-1,j)$= latency time between the j-1 and j[th] node.

Comparison of Client Server model and Mobile Agent based model
In this section the comparison of CS approach and MA approach is being done and performance comparison results of both the approaches are being shown in below table.

Performance Matrix	Client Server Model	Mobile agent based model
$T_{fc}C^{mmt}$ (management cost in terms of network traffic around authenticator)	Directly proportional to number of clients and number of message exchange.	Proportional to size of any information data collected.
$T^m C^{rm}$ (remote interaction time)	Directly proportional to number of messages exchanged on number of clients.	As interaction is local between MA and client it does not increase with increase in number of clients.

International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Parameters	CS	MA (Single user)	MA (Multi user)
Cost of Management	$\sum_{j=1}^{n} \begin{cases} (Srq + Srs) * a * \\ Average\ session\ numbe \\ (Srq + Srs) * b \end{cases}$	$S_{ma} + S_{pr}$	$S_{ma} + \sum_{j=1}^{n} S_{pr}$
Remote Interaction Time	$\sum_{j=1}^{n} \frac{(Srq + Srs)}{BW_j} + 2Lt_j$	$\frac{(S_{ma} + S_{pr})}{BW(j-1,j)} + Lt(j - 1,j)$	$\sum_{j=1}^{n} \frac{(S_{ma} + S_{pr})}{BW(j-1,j)} + Lt(j-1,j)$

Table 1: Comparison of cost and time

4.1 Network Traffic Related Performance

Authentication requires number of message transfer in client server based approach. As the multiple messages exchange increases, the traffic around the authenticator increases to a great fold.

Typical Srq = client server architecture size is around 50 Bytes.

Srq = 50 Bytes

S_{ma} (MA size) is 3 KB= 1024*3 = 3072 Bytes

Traffic generated in CS mode = β times ofS_{ma}, as data generated in the CS approach is much higher than MA approach.

$$(Srq + Srs) = 50 + \beta * S_{ma}$$

Putting these parameters in equation (i)

Case I : Taking β = 8, 50+8*3072 =24626 Bytes

Case II : Taking β = 28, 50+28*3072 = 80066 Byte

No. of Nodes	MA	CS β =8	CS β =28
1	3272	24581	80066
5	4272	123130	445690

10	5272	246260	891380
15	6272	369390	1782760
20	7272	492520	4456900
25	8272	615650	2151650
30	9272	738780	2581980

Table 2: Traffic around authenticator (bytes)

Putting these parameters in equation iii, the management cost for MA based approach at authenticator can be calculated as

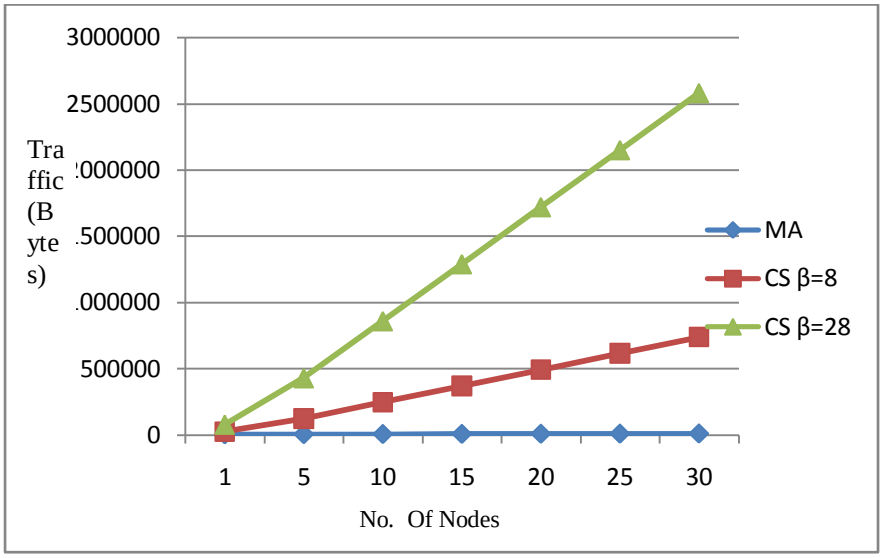
$$C_{ma}^{r} = (Sma + Spr)$$

= (3072+200)=3272 Bytes

It can be analysed from Table that MA based approach really helps in reducing the traffic around the authenticator to a great extent.

The result can be analysed in the graph shown below. Graph shows the results for MA, CS(β=8) and CS(β=28) calculations. In case of MA approach traffic is very less as compared to client server based approach.

Fig 4 : Traffic analysis for Client-Server and Mobile Agent approach



4.2 Remote interaction time Performance

Authentication requires multiple message exchange in client server based approach. As the number of message exchange increases, the time required around the authenticator increases to a great fold.

The turnaround time for authenticator to authenticate the clients will depend on the number of message exchanges taking place between the entities.

- **For client server model:**

$$T^m C^{rm}_{cl-sr} = \sum_{j=1}^{n} \frac{(Srq + Srs)}{BW_j} + 2Lt_j$$

Putting the values:
BW = 10 Kbps = (10 * 10240) bytes/sec = 102400 bytes/sec
Li = 2ms = 0.002 sec

Case 1: Taking μ = 8,(Srq + Srs) = 50+8*3072 = 24626 bytes

$T^m C^{rm}_{cl-sr}$= (24626/102400) + (2 * 0.002) = 0.24448 sec

Case 2: Taking μ = 28,(Srq + Srs) = 50+28*3072 = 86066 bytes

$T^m C^{rm}_{cl-sr}$= (86066/102400) + (2* 0.002) = 0.84448 sec

- **For mobile agent model:**

$$T^m C^{rm}_{ma} = \sum_{j=1}^{n} \frac{(S_{ma} + S_{pr})}{Bw(j-1,j)} + Lt(j-1,j)$$

Putting the values:
$T^m C^{rm}_{ma}$= (3072 + 200) /102400 + 0.002 = 0.033 sec

International Journal of Management, IT & Engineering
Vol. 10 Issue 01, January 2020
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Table 3 Remote Interaction Time for Client-Server and Mobile Agent approach.

	MA	CS β=8	CS β=28
1	0.033	0.24448	0.84448
5	0.165	1.2224	4.2224
10	0.33	2.4448	8.4448
15	0.495	3.6672	12.6672
20	0.66	4.8896	16.8896
25	0.825	6.112	21.112
30	0.99	7.3344	25.3344

Table 3 shows the comparison of remote interaction time in client server approach and mobile agent approach. From the table it is clear that as the number of communicating nodes increases the interaction time around the authenticator increases to a great extent. Graph in figure 5 also helps in clarifying the stigma of remote interaction time. Graph shows that as the number of nodes increases the number of message exchange increases and so it increases the remote interaction time.Graph helps in comparing the mobile agent based approach with the client server approach and shows that traffic is almost constant in case of mobile agent based approach as compared to client server approach, which increases exponentially.

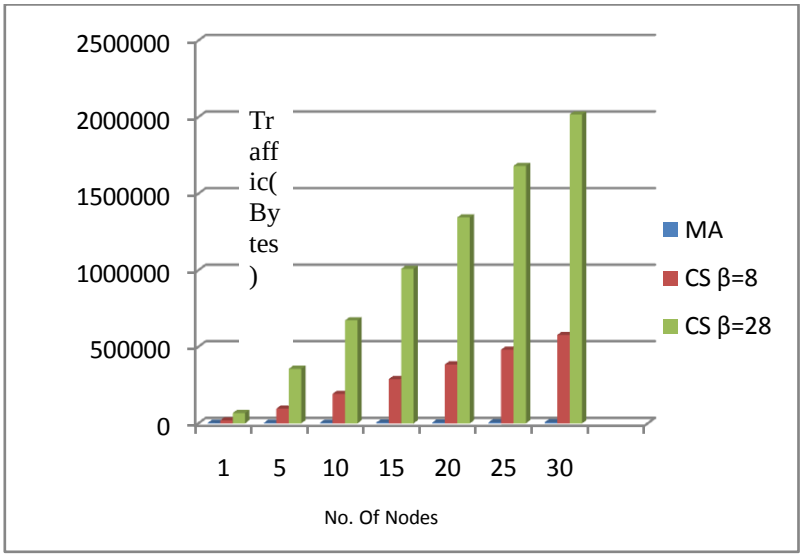


Fig5. Remote Interaction Time for Client-Server and Mobile Agent approach

5Conclusion

In this paper, mobile agent based approach analogous to the client server approach is being proposed. The EAP-LEAP authentication used under the mobile agent approach requires less number of message exchanges, there by reducing the size of data packet, network performance increases as traffic required is very less required just the size of agents, overcome less fault tolerance problem also, centralized management is that the entire network may get out of control after failure of a single manager. Thus using mobile agents, tasks can be easily decentralized. The challenge authentication and the key exchange processes are done on the client end which also reduces the unnecessary exchanges. As discussed in the performance analysis, the mobility of mobile agents reduce bandwidth overloading problems by moving data and context with them thereby saving many repetitive request/response round trips.

References

[1] Miller, Kevin &Mansingh, Gunjan, "Comparing the Use of Mobile Intelligent Agents vs Client Server Approach in a Distributed Mobile Health Application", 2015 Journal of Computers vol. 10, 365-373. 10.17706/jcp.10.6.365-373.

[2] Nimbalkar MV, Nagargoje HM, Pathak GP, VishnudasMB. Mobile agent: Load balanced process migration in Linux environments. Advances in Software Engineering and Systems. 2013;146-150.

[3] Shrouf FA, Turani A, Baker AA, Omri AA. Analysis of mobile agent optimization patterns. British Journal of Applied Science & Technology. 2014;4(12):1841-1857.

[4] U.Kumar and S.Gambhir, "A novel approach for key distribution through fingerprint based authentication using mobile agent," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 3441-3445, 2016.

[5] Atul Mishra, A.K. Sharma, Ashok Madan, "An analysis of management cost for mobile agent based network management model, " International Journal of Computer Applications, pp. 19-28, vol. 42, no.5,2012.

[6] MamtaYadav, PreetiSethi, Dimple Juneja, NareshChauhan" Development of Mobile Agents With Aglets (A Java Based Tool)" International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 4, Special Issue May 2015.

[7] Osunade S, Atanda FA. Analysis of two mobile agent migration patterns. Journal of Mobile Communication. 2008;2(2):64-72.

[8] Muhammad AwaisShibli, SeadMuftic , Alessandro Giambruno, Antonio Lioy, "Security System for Development , Validation and Adoption of Mobile Agents," vol. 56, pp-56-65, Aug 2009.

[9] P. Pacyna and R. Chrabąszcz, "Evaluation of EAP re-authentication protocol," 2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks), Montreal, QC, 2016, pp. 45-49.

[10] Umesh Kumar and Sapna Gambhir, "Mobile agent based framework for wireless authentication", 8th IEEE International Conference on Cloud Computing, Data Science & Engineering (Confluence), **(2018)**, pp. 219-224.

[11] U. Kumar, S. Gambhir, "A literature review of security threats to wireless networks**",** International Journal of Future Generation Communication and Networking, vol.*7*,no.4, **(2014)**, pp. 25-34.

[12] Miller, Kevin &Mansingh, Gunjan, "Comparing the Use of Mobile Intelligent Agents vs Client Server Approach in a Distributed Mobile Health Application", 2015 Journal of Computers vol. 10, 365-373. 10.17706/jcp.10.6.365-373.

[13] SandeepK.Sood, Anil K.Sarje, Kuldip Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," Journal of Network and Computer Applications, vol. 34, pp. 609-618, March 2011.

[14] Higashino M., Takahashi K., Kawamura T., and Sugahara K: ―Mobile agent migration based on code caching‖, 26th International Conference on Advanced Information Networking and Applications Workshops, 651 – 656. 2012.

[15] Mohamed B, Khaoula A, Noreddine G. Communication and migration of an embeddable mobile agent platform supporting runtime code mobility. International Journal of Advanced Computer Science and Applications. 2012;3(1):50-56.

[16] Jie Yu, QingpiPei, "A Multi-server Architecture Authentication Protocol Using Smart Card," Ninth International Conference on Computational Intelligence and Security (2012), pp 511-515, 2013.

[17] Umesh Kumar and Sapna Gambhir," MABFWA: MOBILE AGENT BASED FRAMEWORK FOR WIRELESS AUTHENTICATION", 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 219-224, 2018