IJCTA, 9(40), 2016, pp. 183-189

Key Computation based Secure Handshake Authentication Protocol (KCSHAP) for Secured Distributed Database Access

M. Natarajan* and R. Manimegala**

Abstract : Distributed database which is a group of multiple database interconnected logically and distributed over a computer network, plays a major role in today's business world for storing and retrieving huge business related data. The advantages of implementing distributed database are data replication, low operating costs, faster data access and data processing, but security is still a considerable issue. Hence, secured distributed database architecture based on trusted node is proposed. The architecture contains a special node in a network called a trusted node for each site through which all other nodes will access the database. Trusted node process user requests, combines the results from concerned distributed databases and forward it to the authenticated user. The mechanism adapted by the trusted nodes in order to provide authentication is Key Computation based Secure Handshake Authentication Protocol (KCSHAP). Hence authenticated users can only access the database.

Keywords: Distributed database, security, trusted node, authentication, KCSHAP.

1. INTRODUCTION

The rapid development in the technology of computer network and database system resulted in the growth of distributed database (Mohamed Firdhous, 2011). A distributed database is an assemblage of databases that are distributed and deposited on several computers (sites) within a network. The sites which are involved in the distributed database has the full control over their database in terms of managing the data. The sites may also inter-operate whenever required. A link connection in the database permits the nodes which are local to access the data on a remote database. In order to establish these connections, each database in this system must have a unique name for the global database in the network domain. The name of the global database identifies the database server uniquely in a distributed system. The distributed data are administrated between local and global transactions. A local transactions means that the data can be accessed only by the sites where the transaction located, while a global transaction is that the data have been accessed in multiple site.

The characteristics of trustworthiness and reliability is a primary issue while designing a distributed system. Yet another main issue in the distributed system is the node accessing the database via a remote system that can be attained on the basis of some access rights, policies or authorization semantics (Gulhane, Bodkhe, S, 2015). In a distributed system, the nodes or the users need to be authenticated in both the local as well as the global environment. Here the problem exists that the authenticated system should be distributed or centralized. If the system is centralized, then the authenticator should control and manage the whole users belonging to the system. If the authenticated system is distributed, then the several components of the authenticator required to communicate with each other to authenticate a user.

* Assistant Professor, Department of Computer Science, Thanthai Hans Roever College, Perambalur

** Professor, Department of Computer Science and Engineering, Park College of Engineering and Technology, Kaniur, Coimbatore-641 659 *natrajanm944@gmail.com*

This paper utilizes a distributed authenticated system for secured access in distributed data base management system. Each website consists of a trusted node and this node will authenticate the users belonging to that website based on the Key Computation based Secure Handshake Authentication Protocol (KCSHAP). Trusted node process user requests, combines the results from concerned distributed databases by communicating with other trusted nodes and forward it to the authenticated user.

The rest of the paper is organized as follows: section 2 provides a recent related work done in the secured distributed database management system. The proposed Key Computation based Secure Handshake Authentication Protocol (KCSHAP) based secured distributed management system is explained the section 3. The experimental and results are discussed in section 4. Finally, the section 5 renders the conclusion.

2. RELATED WORKS

(Aruna Kumari, et, al., 2010) proposed a promising solution with trust policies for designing a secured distributed system. In this approach, a service level agreement and a node registry will be performed for the newly joining node. A network authentication protocol namely Kerberos has been used as guarantee the security aspect during client requests for particular services.

A secure concurrency control protocol (SCCP) has been proposed by the authors (Shashi Bhushan, et, al., 2007) based on the time stamp ordering for concurrency control with security maintenance of distributed database management system. This protocol provides the data security during concurrent access to the database. (Hamdi H, et, al., 2009) proposed a framework relied upon security policies for distributed systems. Relied upon a set of abstraction, a modular security policy has been developed in this framework. (Zhang Xing, Hao Wei, 2010) proposes an Intrusion detection system based database security framework by concerning the sensitive data in databases.

(Neera Batra, et, al., 2011) proposed a framework by considering the security issues in distributed data base management system. A polices has been pre established for controlling the users to access the data based on their privileges provided to them. A predefined security policies are replicated at different sites using a multilevel secure database management.

(Li Bai, et, al., 2010) develops a model for a reliable DDBMS to protect the sensitive information. The key idea in this model is to i) contain a (k, n) threshold dependent secret sharing scheme to provide security from being information stolen. ii) An effective distributed database management design is incorporated to improve the system performance and reduce interfered accesses contentions and iii) The private information storage structures has been integrated to minimize communication overhead and enhance system robustness.

A fragment allocation scheme called S-FAS has been investigated by the authors (Yun Tian, et, al., 2011) to enhance security of a distributed system where storage sites have a numerous vulnerabilities. A file fragmentation is integrated with the secret sharing method in the S-FAS approach. Based on the vulnerability characteristics the storage sites have been categorized into a various server types. The fragments of the given file are allocated by the S-FAS to various types of nodes. The confidentiality of data has been safeguarded by allocating the fragments in multiple storage nodes.

(Min Zhang, et, al., 2008) proposes an architecture for the secured storage system, where a layer called management nodes are added between the storage nodes and users. A secure and reliable storage services are offered by the architecture while the servers suffer from different types of intrusion. The scope of trust is reduced by the architecture from a a large scale of servers to a some managed nodes and a flexible access control is provided to support the integrity, privacy and availability of data while safeguards the efficiency, flexibility and scalability.

(Feng Shen, et, al., 2010) proposes a Multi-coefficient Secret Sharing by modifying the Shamir's secret sharing scheme for Secure and Reliable Data Storage. The secret size should not too long in the shamir's secret key sharing, to overcome this the multiple coefficients are utilized in polynomials hence that secret sharing sharing can able to useful for managing data in distributed environments.

(Xinyi Huang , et, al., 2011) proposes a generic and secure framework for secure distributed system. The authentication is relied upon three factors such as password, biometrics and smart cards. The client privacy is ensured in distributed system by protecting the resources from the unauthorized users.

A six bit encryption scheme is utilized by the authors (Gurkamal Bhullar, et, al., 2014) to enhance the security in a distributed database. This algorithm improves the security with concurrency control in a distributed database. While in traffic the query redirection factor is used to control the concurrency.

3. KEY COMPUTATION BASED SECURE HANDSHAKE AUTHENTICATION PROTOCOL (KCSHAP) BASED SECURED DISTRIBUTED MANAGEMENT SYSTEM

A Secure database model is considered in this system where the global database is partitioned into a collection of local databases and distributed over N sites connected via a network. A independent processor has been equipped with each site, which has been connected through a secured communication link to other sites. Each site consists of a n number of client nodes and a trusted node. The trusted node processes the user requests from the client nodes. The combines the results from concerned distributed databases and forward it to the authenticated user. The trusted node authenticates the user based on the Key Computation based Secure Handshake Authentication Protocol (KCSHAP). The architecture of the proposed system model has been shown in the figure 1.

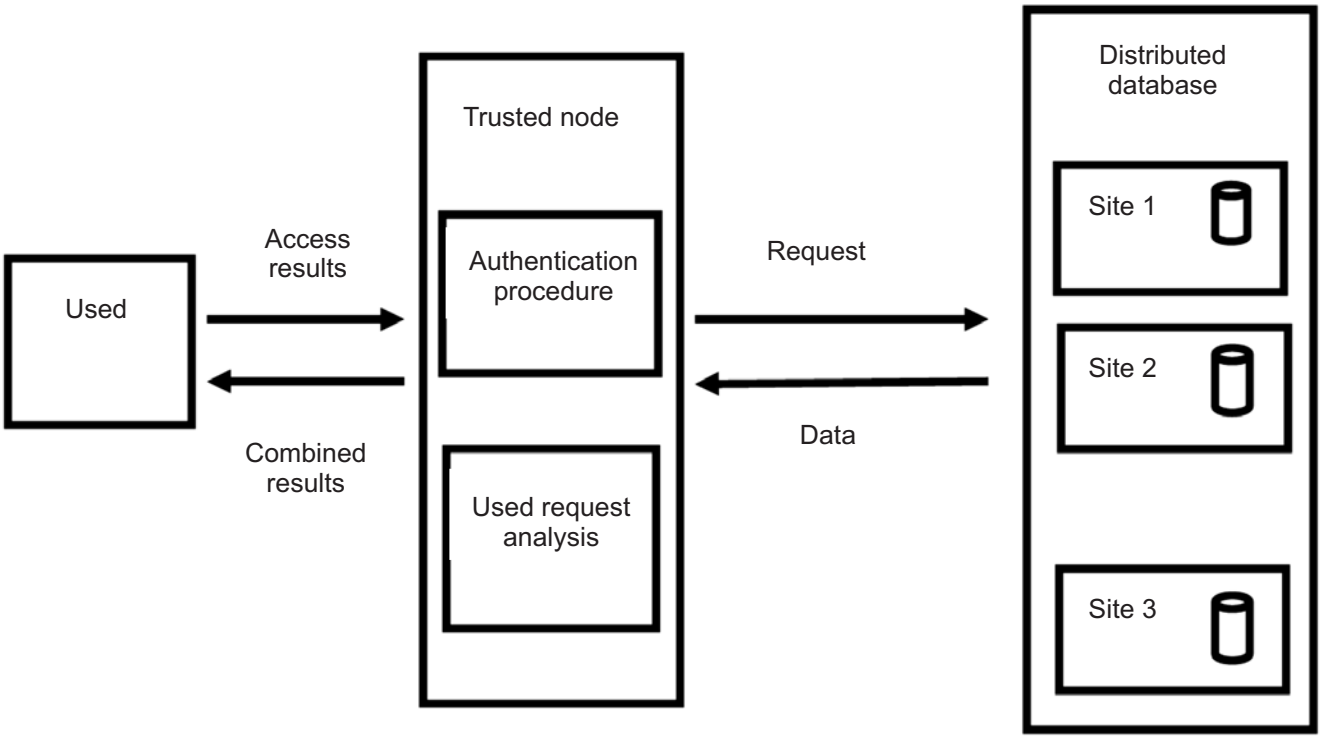

Figure 1: Architecture of proposed Key Computation based Secure Handshake Authentication Protocol (KCSHAP) based secured distributed management system

3.1. Key Computation for the Clients Node Belonging to the Site

Initially, when a new node registered with the site is termed as client node. The pseudo code for the key computation of each client node is given in the algorithm 1. The trusted node belonging to the site will assign the private security number PSN (prime number) for that node, which request to join in the group. The size of the prime number may be 1024 bits. A random group key GK > PSNi is selected by the trusted node for all i members belonging to the site and computes the message (a_i, b_i) pairs by using the following equation

$$a_i = \frac{GK}{PSK_i} \quad (1)$$

$$b_i = GK \bmod PSK_i \quad (2)$$

Finally the trusted node computes a password *pwd* for that node belonging to the site by using the following equation

$$pwd = a_i \times PSN_i + b_i \quad (3)$$

A random salt s is assigned to a node and a verifier *v* is computed for that node based on the following equation

$$v = gx \quad (4)$$

Where *g* is a generator of the multiplicative group, $x = h(s, pwd)$ is the long term private key and *h* is the cryptographic hash function.

Now the trusted node stores the *a*, *b*, *v* and *s* for verifying the node at the time of database access request.

Algorithm 1: Pseudocode for key computation for each client node belonging to the website

Step 1: For *i* to *m* websites in distributed database management system

Step 2: Assign a trusted node to the website *i*

$$TN_i \in WS_i$$

Step 3: For *i* to n nodes in the websites

Step 4: Assign Private Security Number PSN for the nodei,which is a prime number of size 1024 bts

Step 5: end for

Step 6: A group key GK is randomly selected and it should satisfy

$$GK > \forall PSN \text{ assigned to the nodes}$$

Step 7: For *i* to n nodes in the websites

Step 8: Compute the message pair (a_i, b_i) using the equation 1 and 2

Step 9: Compute the password pwd using the equation 3

Step 10: Assign a random salt *s* and compute a verifier *v* using the 4

Step 11: end for

Step 12: end for

3.2. Authentication based on Trusted Node

When a node requests the trusted node to access the database locally or globally, the node will send its IP address to it. The trusted node looks the verifier v and the salt s for that IP address.The trusted node sends a challenge to the node, which composes of an identifier ID, random number r and salt *s*. The challenge is accepted by the node and it computes the long term private key $x = h(s, pwd)$. After that it computes a Response R based on the following equation

$$R = g^{ab + brx} \quad (5)$$

Where *a* and *b* are the computed message pair at the time of registration in the site, no member other than that node can get the response R using the hidden pwd and the key pairs. The response to the challenge is sent to the trusted node with an encrypted ID and the trusted node checks the response against its own computations of the expected R value. If the value matches, then the trusted node acknowledges the authentication and sends a success message to the node and establishes the link for the corresponding database access request. Otherwise the node is not authenticated to access the database. The sequence diagram for the authentication procedure based on the trusted node at the time of the database access request has been given in the figure 2

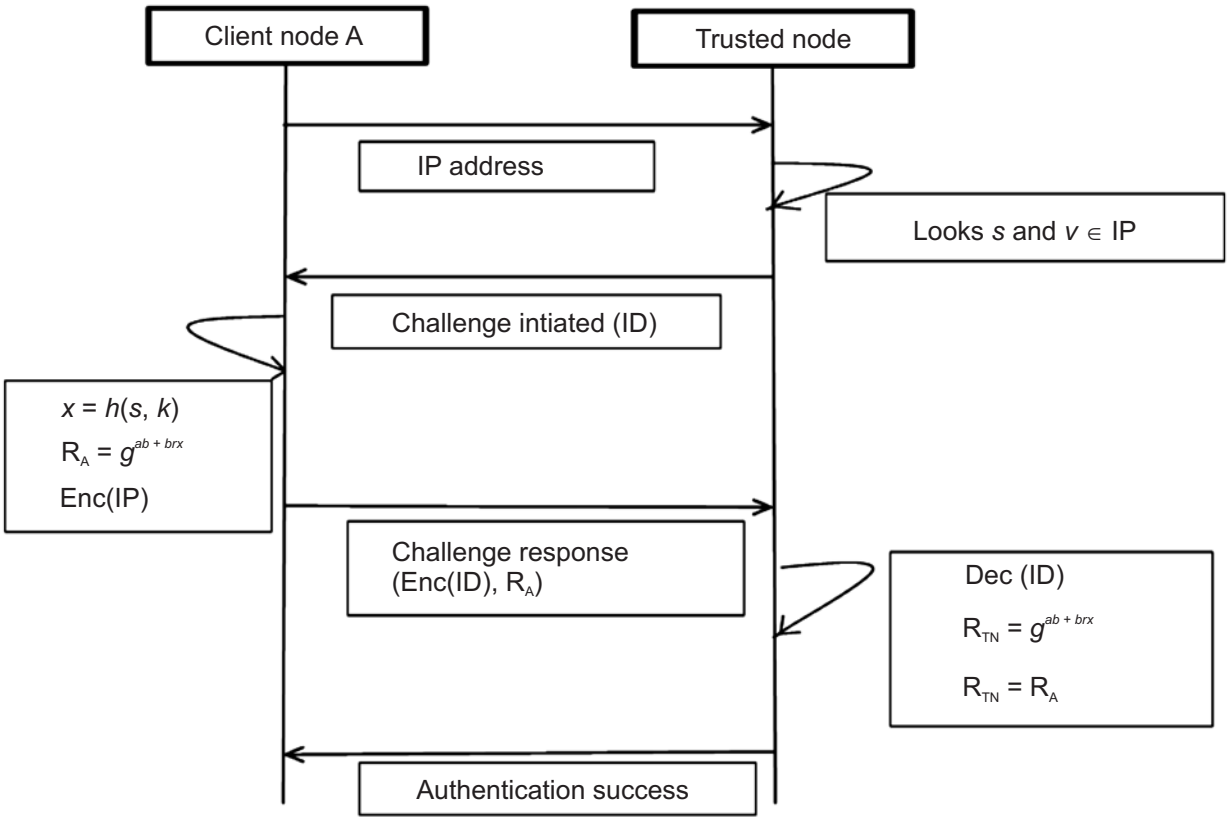

Figure 2: Sequence diagram for the authentication procedure based on the trusted node

4. EXPERIMENTS AND RESULTS

The proposed KCSHAP is evaluated by using the Network Attached Storage (NAS), which one of the popular distributed storage system. To transfer the data across multiple devices or nodes, the NAS uses the TCP/IP protocol. The experiment is carried out of 6 sites and 50 nodes in the network. The nodes and the site are interconnected by a Gigabit Ethernet. The nodes in the sites have CPUs ranging from dual-core 2.80GHz Pentium Ds to 2.83GHz quad-core Xeons, and run Fedora 10 with Linux Kernel 2.6.27. The access control list has been created to restrict the user from accessing the data at different levels. The proposed system is compared with the existing Multi-coefficient Secret Sharing (MCSS) in terms of computation complexity and number of access granted for the request. The computational problem is solved with some complexity can be defined in terms of power utilized with respect to time and it can be given as follows

$$\text{Computation complexity} = \text{Power utilized} \times \text{time} \quad (6)$$

Figure 3 shows the computation complexity with respect to a number of database access request. In a network, the number of access request increases, the authentication time to permit such amount of access will increase the computational complexity. The proposed KCSHAP is a simple procedure when compared to the MCSS, to authenticate the users. Furthermore, the distributed trusted node in KCSHAP to authenticate the users without the central authority will make the authentication procedure fast to the requested client nodes. For 50 numbers of an access request, the KCSHAP reaches computation complexity of 1500 Js, while the MCSS reaches computation complexity of 2000 Js.

Figure 4 shows the Security analysis for the KCSHAP and MCSS in terms of access granted with respect to access request from n number of users. In KCSHAP, the computed message pair a and b and the password pwd at the time of key computation is hard to break by the hackers. The access request is not from the concerned user will be denied for the particular access request during the authentication procedure. For

37 access requests from the 50 users in the system, the KCSHAP granted 30 access requests by denying 7 requests, while the MCSS granted 33 access requests by denying 4. It is clear that the proposed KCSHAP performs better security than the existing MCSS.

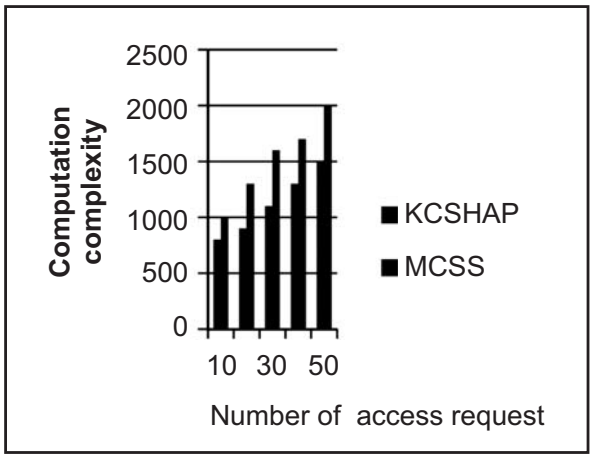

Figure 3: Computation complexity with respect to number of access request

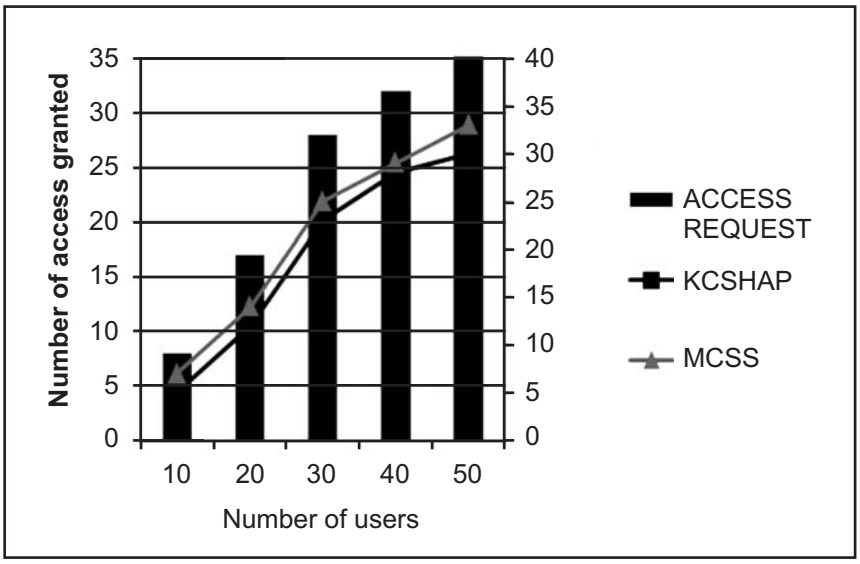

Figure 4: Security analysis for the KCSHAP and MCSS in terms of access granted with respect to access request from n number of users

5. CONCLUSION

This paper proposes a Key Computation based Secure Handshake Authentication Protocol (KCSHAP) for secured distributed database management system. This system employs a trusted node in each site to manage the user access to the database locally or globally. The trusted node belongs to the site will generate a password, random salt and a long term private key for the client nodes connected to a site using a key computation algorithm. If a client node initiates an access request to the specific database, the trusted node presents a challenge to that node in terms of an identifier, salt and a random number. The client node accepts the challenge and gives a response by some computation. The trusted node does the same computation to verify the response given by that node. If the response is same, the node is authenticated to access the database. Then the Trusted node process user requests, combines the results from concerned distributed databases and forward it to the authenticated user. The experimental results show that the proposed protocol performs better than the existing Multi-coefficient Secret Sharing (MCSS) scheme in terms of computation complexity and security.

6. REFERENCES

1. Aruna Kumari, Shakti Mishra, D.S. Kushwaha, "A New Collaborative Trust Enhanced Security Model for Distributed System", International Journal of Computer Applications, vol 1, no 26, 2010.
2. Shashi Bhushan, R. B. Patel and Mayank Dave, " A Secure Time-Stamp Based Concurrency Control Protocol For Distributed Databases", Journal of Computer Science, vol 3, issue 7, pp 561-565, 2007.
3. Neera Batra, Manpreet Singh, "Multilevel Policy Based Security in Distributed Database", Advances in Computing and Communications, Springer, vol 190, pp 572-580, 2011.
4. Hamdi, H, de Bordeaux, Mosbah, M, "A DSL Framework for Policy-Based Security of Distributed Systems", International Conference on Secure Software Integration and Reliability Improvement, Shanghai, IEEE, pp 150 – 158, 2009.
5. Zhang Xing, Hao Wei, "The structure design of database security monitoring system based on IDS", International Conference on Computer Engineering and Technology (ICCET), vol 3, pp 450-453, 2010.
6. Li Bai, Biswas, S, Ferrese, F, "Design of a Reliable Distributed Secure Database System", International Conference on Networking, Architecture and Storage (NAS), Macau, IEEE, pp 91 – 99, 2010.
7. Yun Tian, Shu Yin, Jiong Xie, Ji Zhang, Xiao Qin, Alghamdi, M.I, Meikang Qiu, Yiming Yang, "Secure Fragment Allocation in a Distributed Storage System with Heterogeneous Vulnerabilities", International Conference on Networking, Architecture and Storage (NAS), Dalian, Liaoning, IEEE, pp 170 – 179, 2011.
8. Min Zhang, Desheng Zhang, Hequn Xian, Chi Chen, Dengguo Feng, "Towards A Secure Distribute Storage System", International Conference on Advanced Communication Technology, Gangwon-Do, IEEE, vol 3, pp 1612 – 1617, 2008.
9. Feng Shen, Chonglei Mei, Hai Jiang, Zhiqian Xu, "Towards Secure and Reliable Data Storage with Multi-coefficient Secret Sharing", International Conference on Computer and Information Technology (CIT), Bradford, IEEE, pp 797 – 802, 2010.
10. Gurkamal Bhullar, Navneet Kaur, "Enhancing Security and Concurrency in Distributed Database with 6 Bit Encryption Algorithm", International Journal of Computer Science and Information Technologies, vol 5, issue 3, pp 2718-2722, 2014.
11. Mohamed Firdhous, "Implementation of Security in Distributed Systems – A Comparative Study", International Journal of Computer Information Systems,vol. 2, issue 2, 2011.
12. Xinyi Huang , Yang Xiang , Chonka, A, Jianying Zhou, Deng, R.H, "A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems", Transactions on Parallel and Distributed Systems, IEEE, vol 22, issue 8, 2011.
13. Gulhane, Bodkhe, S, "DDAS using Kerberos with Adaptive Huffman Coding to enhance data retrieval speed and security", International Conference on Pervasive Computing (ICPC), Pune, IEEE, pp 1-6, 2015.