



## Analysis and Implementation of Kerberos Protocol in Hybrid Cloud Computing Environments

Turkan A. Khaleel  <sup>a\*</sup>

<sup>a</sup> Department of Computer Engineering, Engineering College, University of Mosul, Nineveh, Iraq,

[turkan@uomosul.edu.iq](mailto:turkan@uomosul.edu.iq)

\*Corresponding author.

Submitted: 14/04/2020

Accepted: 16/08/2020

Published: 25/03/2021

### KEYWORDS

Kerberos protocol,  
Hybrid Cloud  
Computing; Network  
Security, and  
Authentication.

### ABSTRACT

*The concept of cloud computing has recently changed how hardware, software, and information are handled. However, security challenges and credibility requirements have never changed and may have increased. Protecting cloud computing and providing security for its resources and users is one of the critical challenges. As a result, most users are afraid to use their resources, because many security problems must be met. For example, authentication and reliability are major security constraints and must be provided in a cloud computing environment. There is a wide range of authentication protocols in use, but the researcher has recommended the Kerberos protocol to represent and test it in a complex environment such as a mixed cloud environment.*

*A model has been developed to implement Kerberos authentication in a hybrid cloud computing environment to securely access the cloud computing services provided. This model is represented using the OPNET Modeler 14.5 simulation system. The network efficiency was measured before and after the hacker. Findings presented in this research are supporting the ability of the Kerberos protocol to prevent illegal access to cloud computing services, whether from within the private cloud or the public cloud. While maintaining the efficient performance of the network.*

**How to cite this article:** T. A. Khaleel, "Analysis and Implementation of Kerberos Protocol in Hybrid Cloud Computing Environments," Engineering and Technology Journal, Vol. 39, Part B, No. 01, pp. 41-52, 2021.

DOI: <https://doi.org/10.30684/etj.v39i1B.1675>

This is an open access article under the CC BY 4.0 license <http://creativecommons.org/licenses/by/4.0>

## 1. INTRODUCTION

A hybrid cloud is an important type of cloud. This type provides the integration of resources between private and public clouds. It offers users to expand their internal infrastructure horizontally in the public cloud to improve performance and reduce investment costs [1]. A hybrid cloud is a type of cloud computing that provides many cloud services without constraining local resources because it often includes different types of cloud that may include a private cloud with a public cloud intended to facilitate the user's work. With recent changes in computing, management, and increasing use, hybrid cloud computing may give everyone assessed and use of resources and expand their local

infrastructure by linking to local and private clouds. While providing as much as possible confidentiality and reliability and protecting its users from any violations. Additionally, they provide their services without allowing third-party data centers to access their data in full. Special drawing sources and systems are more protected than other types of clouds. But public computing may give organizations more flexibility and power to perform their tasks much better than private cloud. Especially if there is protection such as a firewall to protect the sources of information and prevent intrusion and unauthorized access [2].

In short, the hybrid cloud can be expressed as a strategy for sharing resources by creating cloud computing that combines different cloud environments connected. These cloud environments can be either a public or private cloud or a virtual infrastructure [3]. The hybrid cloud has many advantages, and although it is one of the most stable cloud environments, it still faces some challenges. One of the most important challenges is the issue of authentication and confidentiality [4]. In hybrid cloud computing, protecting information security is one of the most important issues to address. Among various modern security technologies used Maintaining privacy and protecting a connection method During the cloud, authentication technology is a vital issue. User authentication has become a vital problem in cloud computing because of the urgent need to protect critical information in cloud service providers [3][4].

Although there are many studies in this field, there are no systems capable of evaluating and testing the information security program that applies to hybrid cloud computing. In the study [5], the establishment of a testing and evaluation system consists of three parts: the test environment, the evaluation technology, and the evaluation service. In the cloud computing environment, many authentication protocols have been proposed such as (Choudhury et al. [6], Nayak, et al., [7], Emam [8], Banyal, et al., [9], Khan and Akbar [10], Abdul et al., [11]) and many efforts have been devoted to improving cloud computing security. Among these proposed authentication protocols, Kerberos (Zhao et al. [12]).

In this article, a hybrid cloud computing model with Kerberos protocol authentication is studied and proposed to provide secure login and give users more confidence when using this type of cloud. This proposed model can be implemented in practice with available capabilities as well as protection for hybrid cloud computing resources. While maintaining the efficiency and performance of the network and the possibility of providing resources and systems for all users. Kerberos verifications provide authentication for each client as well as authorizing two authorized users to use hybrid cloud resources. It acts as a trusted third party between hybrid cloud servers and clients and thus allows secure access to services. The remainder of this paper is organized as follows: Section 2, provided an overview of the authentication and confidentiality of the Kerberos Protocol. Section 3 discussed the proposed work which includes implementation, and the model description. Section 4 discusses the results of the analysis and simulation. Conclusion and future work are presented in section 5.

## 2. OVERVIEW OF THE KERBEROS SECURITY

In the simplest definition, Kerberos is an authentication protocol that enables it to serve requests over insecure networks and unreliable requests as on the Internet environment. In the Kerberos authentication protocol, we have three headers: Client Centre, Server, and Key Distribution (KDC). KDC is a Kerberos third-party authentication service. KDC is an intermediary between the clients and the server with confidence requests provided by the KDC. KDC provides two basic services: (Authentication Service (AS) and Ticketing Service (TGS)). The KDC authentication mechanism uses a shared secret key that allows the transmission of packets that travel on the network safely. This means protecting and preventing replay attacks [13]. In general, the work of this protocol is based on three phases: Phase 1: At this stage, the client must authenticate and request a ticket from the Kerberos protocol. At this stage, Kerberos will present the Ticket Granting Ticket (TGT) and session key. This ticket is used to request a ticket to allow access to the server and use the resources and services provided by the server. The session key is used to communicate between clients and the server. Phase 2: The client uses tickets from Phase1 to request a ticket from TGS. Stage 3: The client uses the ticket to connect to the intended server. In this case, the client sends a request in plain text to the Authentication Administrator (AS) server so that it can obtain a card to subsequently communicate with TS. The request contains the login name as TGS. AS gets the authentication name and is logged in [13].

The Kerberos service generally consists of a client-server architecture that provides secure transactions across networks. Thus, it provides a strong authentication service to the user while

maintaining the integrity and privacy of the user. The identity of the sender and receiver is validated and provides authentication for the parties within the network. It also provides additional services such as data integrity checking and encryption during transmission according to equation (1) [14]. As shown in Figure 1.

$$D_{key}(E_{key}(Message)) = Message \quad (1)$$

Where E, is the encryption process and D, the decryption process. In these two processes (encryption and decryption) the same key is used. With this key, the sent ciphertext is decrypted. In Kerberos reliability, there are two types of keys used: symmetric short-term keys and symmetric long-term keys.

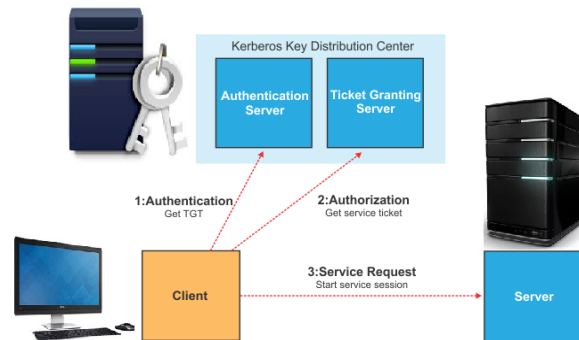


Figure 1: Kerberos, client-server architecture.

### I. Kerberos Authentication Steps

Kerberos is a ticket-based authentication protocol for clients. With these tickets, users are allowed to connect to services over a potentially unsafe network. The following are the most important steps to implement Kerberos authentication, which should be implemented to get the client to obtain a secured request from the server before using the server resources. At the same time, it represents how Kerberos works briefly [14].

#### Step 1: Sign in

In this step, the client creates its account consisting of the user name and password. Later in normal cases, the client is required to enter the password only. This is what we notice in step 5. This password will then be converted to a secret client-key.

#### Step 2: Request the client to the server granting the ticket (TG).

The client sends a text message request to the authentication server (KDC). The message contains: (username; type of service required) In this case, TGS is given and includes the network address and TG life. At this point, confidential information consisting of the secret client key and password has not yet been sent.

#### Step 3: The server verifies that the client profile exists.

After the server receives the message and checks whether the client name exists in the Key Distribution Centre (KDC). The purpose of this step is not to verify the credentials, but simply to verify the client profile. The server made sure there was no problem.

#### Step 4: Again the server sends TG to the client.

In this step, the server randomly generates a key called a session key. This shared key will be used between clients and TGS. The authentication server sends two messages (A, B) to the client. Message A is encrypted using the client's secret key. The client's secret key is not transferred but is retrieved from the password stored in the server database. All these processes will occur on the server-side. Message A contains the following information: (TGS name, TGS session key, a timestamp, and lifetimes). The letter B, which is a TG encrypted and using the secret key TGS.

Message B contains: (client name, TGS name, network address, TGS session key, a timestamp, and lifetime).

**Step 5: The server asks the client to enter its password.**

In this step, the client receives messages asking the client to enter its password. To have an account that was set up in the previous step 1. The password is then converted (hashed) to the secret client-key. At the same time, this key is also created on the server-side.

**Step 6: The client obtains the TGS session key.**

In this step, the client uses the secret client key to decrypt the message A. The client gives the TGS session key. For message B, the client cannot do anything at this time, as it will be encrypted using the TGS secret key (which is only available on the server-side). Locally the encrypted TG for credentials is stored in the cache.

**Step 7: The client asked the server to access the resources.**

The client configures two messages (C and D) to be sent to the server. Message C is an unencrypted text message that will be sent to the server and contains: (the service the client wants to access and time to live and the encrypted message B included in this unencrypted message). Message D is an authentication, message that is encrypted using the TGS session key and contains: (client name and timestamp).

**Step 8: The server checks the availability of the service.**

In this step, the KDC server first verifies the service requested by the client. If the service is available and available, continue or send a warning message.

**Step 9: The server checks the request.**

In this step, the server extracts the content of message B (TGT) from message C, decrypts that message B (TGT) with its secret TGS key. The server then gives the TGS session key. This key will be shared between the client and the server. With this TGS session key, the server can now decode Message D.

**Step 10: Create a service session key from the server.**

The server generates a service session key as well as a random key to encrypt and send two messages (F and E) to the client. Whereas, the letter E: is the service ticket and encrypted with the service secret key and contains: (client name; service name; Timestamp; Client network address; Times of life; Service session key). While the message F: is a message encoded with a TGS session key that contains: (The name of the service; Timestamp; Times of life; Service session key).

**Step 11: The client receives the service session key.**

Because the client has the temporary TGS session key resulting from the previous steps, they can now decrypt message F with the service session key. But it cannot decrypt a service card (or E-message) because it is encrypted using the secret service key.

**Step 12: Client access to the service.**

This step enables the client to access the service. Two messages (G and H) are sent. Message G is encrypted using the service session key to achieve new authentication. Includes (name and timestamp). The letter H is the same as the previously received (E) message, which is still encrypted with the secret service key.

**Step 13: The server receives the request.**

When the server receives the request, it decrypts message H (similar to Message E) using a secret service key. The service session key (which is stored in message H) is used. The server then uses the newly acquired service session key to decrypt the authentication message to G.

**Step 14: Check service available on request.**

Similar to step 9, the server verifies the service request sent by the client.

**Step 15: Verify the identity of the user requesting the service.**

The server will confirm the identity of the client based on the authentication message sent by the client. This message (I) is encrypted using the service provider session key that contains (ID and Timestamp).

**Step 16: Receive the client's confirmation.**

In this step, the client receives confirmation that it is the authentication message (I) and decrypts it by using a service session key that was obtained in step 11). To allow the customer to know the identity of the service. And check whether the timestamp is valid.

**Step 17: The client access the service.**

Finally, the client has been authenticated and verified. Here the customer can access the service. Kerberos is the protocol that eliminates the need to own and secure private keys. Allow users to authenticate themselves without transferring their passwords on the network. Whereas, X.509 certificates rely on the concept of asymmetric encryption as the private key is owned by the certification authority and the public. The key is distributed to the user who wants to own the certificate from the certification authority. The problem with this protocol is the maintenance of a database of these keys [15].

**II. Mutual authentication**

In mutual authentication, the server and client must verify their identities to each other before performing application functions. Where neither of the two parties can ever carry out operations with each other unless the identity is verified. In this application, special emphasis was placed on the effect of adding security to hybrid cloud computing and service delivery performance. We assume P5\_AS1 Talk, consists of two distinct parts: (1) Authentication Session key discovery and (2) encrypted message exchange. The second A version of its app, which we call P5\_AS1 Talk, only performs regular text exchanges. Authentication and Session Key Discovery part of Secure P5\_AS1 Talk This model works as shown below. We will refer to the message Msg encrypted with Key such as Key { Msg }.

1. Client (P5) sends a message to the Key Distribution Center (KDC), a trusted third-party authority, requesting a session key for P5 communication with Application Server1(AS1).
2. KDC generates a session key,  $K_{P5\_AS1}$ , for communication between P5 and AS1.
3. KDC sends a message  $K_{P5\_Pu} \{ K_{P5\_AS1}, T_{AS1} \}$  to P5, where:
  - a.  $K_{P5\_Pu}$ : Client P5's public key.
  - b.  $T_{AS1}$ : ticket generated by KDC to be sent to AS1.  $T_{AS1} = K_{AS1\_Pu} \{ K_{P5\_AS1}, "P5" \}$ .
  - c.  $K_{AS1\_Pu}$ : AS1's public key.
  - d. "P5": Client P5's identity.
6. Client P5 receives the above message from the KDC and decrypts it using his private key, providing P5 with  $K_{P5\_AS1}$  and  $T_{AS1}$ .
5. Client P5 sends a message  $\{ T_{AS1}, K_{P5\_AS1} \{ S_n \} \}$  to AS1 where  $S_n$  is some secret number.
6. AS1 receives the message from P5 and decrypts  $T_{AS1}$  using his private key, thereby obtaining  $K_{P5\_AS1}$ .
7. AS1 uses the retrieved key  $K_{P5\_AS1}$  to decrypt P5's secret number  $S_n$ . This proves P5's identity because only someone who knows P5's private key could have obtained  $K_{P5\_AS1}$  and the  $T_{AS1}$  ticket containing P5's identified in step 4.
8. AS1 encrypts and sends P5 a message  $K_{P5\_AS1} \{ S_{n+1} \}$ . AS1 encrypts the value  $S_{n+1}$  to protect against a replay attack.
9. P5 decrypts AS1's message using the session key  $K_{P5\_AS1}$ , verifying AS1's identity, because only AS1 could have known the value  $S_n$  and the session key  $K_{P5\_AS1}$ .

The steps above can be represented and summarized in Figure 2. as shown below.

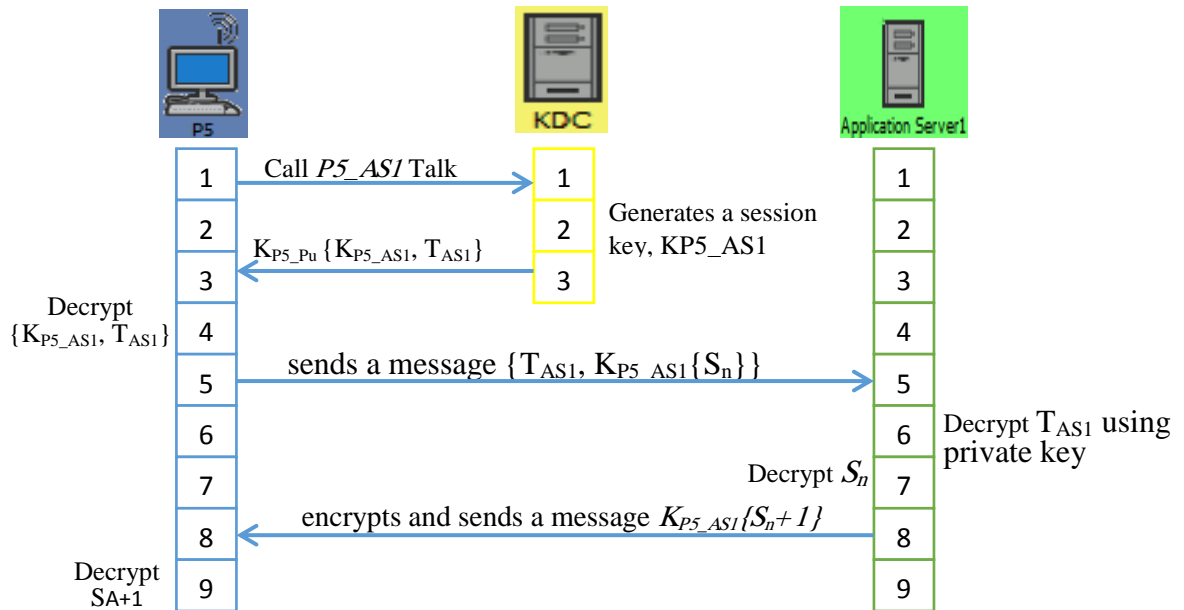


Figure 2: Kerberos Authentication

### 3. IMPLEMENTATION AND MODEL DESCRIPTION

This section describes a hybrid cloud computing model that uses the Kerberos protocol to develop authentication for the use of cloud services provided by servers. The hybrid cloud computing model is designed using OPNET Modeler 14.5 as shown in Figure 3. The hybrid cloud (as shown in the yellow area) consists of a private cloud (as shown in the blue area) as well as general cloud computing (e.g., shown in the green area). The attributes of each node are detailed in Table I. This table shows the types of devices used and their characteristics. In the research, a very simple model was assumed for one private cloud and only one public cloud.

In this work, the cloud provides a wide range of services, including database applications and web services. In the scenario shown in Figure 3. A custom application and connectivity between multiple nodes are presented and are not limited to a single client-server session. In this scenario, Kerberos requirements are prepared as described in Section 2.1. For all routers connected to the hybrid cloud as shown in Figure 4. In this scenario, it is assumed that the P5 user exists within the private cloud domain and needs access services available in the public cloud. The KDC server is in the hybrid cloud responsible for the PR1, PR2, and P1\_AP routers configured with the Kerberos protocol to give, credibility to access services.

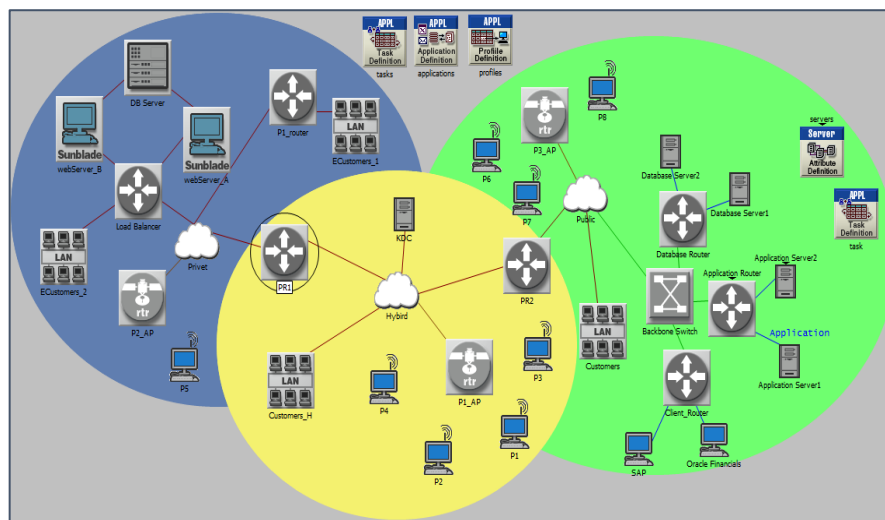


Figure 3: Kerberos Authentication

Phase Name	Start Phase After	Source	Destination	Source
Transaction #1	Application Starts	Originating Source	Application Server	(...)
Transaction #2	Previous Phase Ends	Application Server	Not Applicable	(...)
Transaction #3	Previous Phase Ends	Application Server	Database Server	(...)
Transaction #4	Previous Phase Ends	Database Server	Application Server	(...)
Transaction #5	Previous Phase Ends	Application Server	Originating Source	(...)

Phase Name	Start Phase After	Source	Destination	Source->
Talk to KDC	Application Starts	PR2	KDC	(...)
Talk to PR1	Previous Phase Ends	PR2	PR1	(...)
Verify PR1	Previous Phase Ends	PR2	Not Applicable	(...)

e-commerce	(...)	Refresh After Task
Kerberos Authentication	(...)	Refresh After Task
Secure Transmission	(...)	Refresh After Task
Plain Text Transmission	(...)	Refresh After Task
Custom_Task	(...)	Refresh After Task

Figure 4: Tasks Setting





































[-] Security	
[-] Router Security Parameters	
[-] AAA Parameters	Not Configured
[-] Kerberos Parameters	(...)
Local Realm	KDC
[-] Server Information	(...)
Number of Rows	1
[-] KDC	
Realm Name	KDC
Host Identifier	1
Port Number	88
[-] Host-Realm Mapping	(...)
Number of Rows	1
[-] P5	
Host Name	P5
Realm Name	Application Server1

Figure 5: Kerberos router setting

#### 4. ANALYSIS AND SIMULATION RESULTS

This section includes displaying and analyzing some results obtained when applying a Kerberos protocol in the hybrid cloud model and the simulation time was one hour. In this model, two scenarios were implemented: the first is that the client P5 within the private cloud and wants to access the public cloud to take advantage of the service provided by Application Server1. Client P5 as in the Kerberos dialogue sends Ticket Request to the KDC server. Which sends service Tickets to client P5. The client P5 then sends a request to the cloud to obtain the services provided by it. The Ticket is then checked by the server if it is valid. The server sends the responses and services are provided to this client as shown in the results in Figures (6-7).

**TABLE I: Attributes of node models**

Device Types	Model
External AS	atm16_eth64_fr16_sl64_cloud_adv
 Hybrid	atm16_eth64_fr16_sl64_cloud_adv
 Privet	atm16_eth64_fr16_sl64_cloud_adv
 Public	
<b>LANs</b>	
 Customers	llm_lan_adv
 Customers_H	llm_lan_adv
 ECustomers_1	llm_lan_adv
 ECustomers_2	
<b>Routers</b>	
 Application Router	atm4_ethernet2_gtwy
 Client_Router	atm4_ethernet2_slip8_gtwy_adv_59_upgrade
 Database Router	atm4_ethernet2_gtwy
 Load Balancer	load_balancer_e16
 P1_AP	wlan_ethernet_slip4_adv
 P1_router	CS_3640_4s_e5_fe1_tr1_sl6_adv_13_upgrade
 P2_AP	wlan_ethernet_slip4_adv
 P3_AP	wlan_ethernet_slip4_adv
 PR1	WLAN-NY_hot_spots_ISP_router_84_upgrade
 PR2	WLAN-NY_hot_spots_ISP_router_84_upgrade
<b>Servers</b>	
 Application Server1	ethernet_server_adv
 Application Server2	ethernet_server_adv
 Database Server1	ethernet_server_adv
 Database Server2	ethernet_server_adv
 DB Server	Sun_Enterprise_4500_14CPU
 KDC	ethernet_server_adv
 webServer_A	Sun_Blade_1000_Model_2900
 webServer_B	Sun_Blade_1000_Model_2900
<b>Switches</b>	
 Backbone Switch	atm8_crossconn
<b>Workstations</b>	
 Oracle Financials	ethernet_wkstn_adv
 P1	wlan_wkstn
 P2	wlan_wkstn
 P3	wlan_wkstn
 P4	wlan_wkstn
 P5	wlan_wkstn
 P6	wlan_wkstn
 P7	wlan_wkstn
 P8	wlan_wkstn
 SAP	ethernet_wkstn_adv



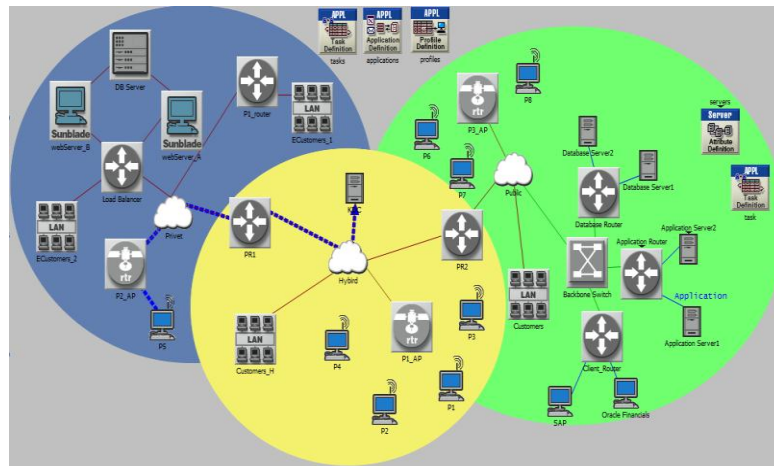


Figure 6: Client (P5) Kerberos request path

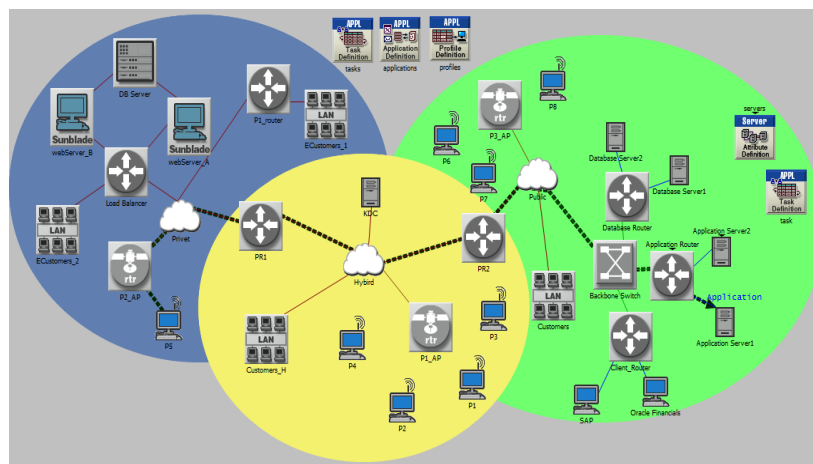


Figure 7: client (P5) Kerberos response path

The second scenario in this model is that the client P5 is a hacker and illegal who wants to access the service provided by the hybrid cloud. After applying the Kerberos protocol, we note the response of the system to Kerberos' work, as this node is not allowed to access the service because it is illegal and as shown in Figure 8. This scenario shows the ability to access all nodes for all the hybrid clouds normally with the P5\_Hacker disconnected from the network.

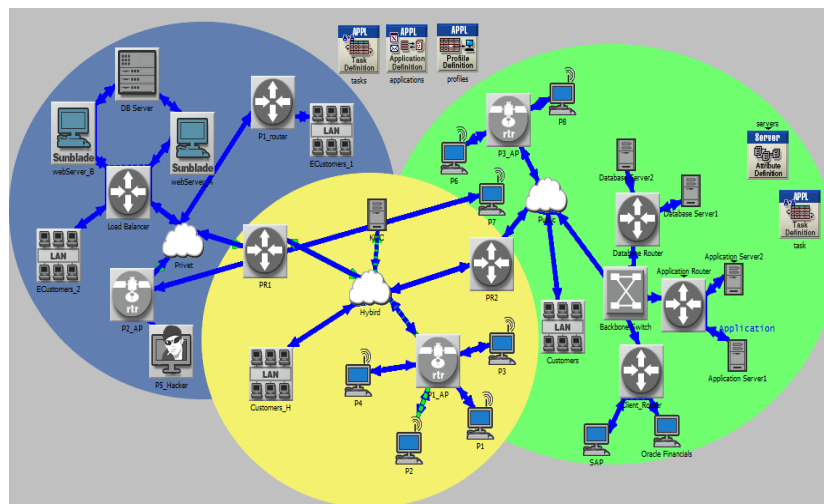


Figure 8: Client (P5\_Hacker ) Kerberos response path

Figures (9-11) show some statistics obtained when implementing the model designed when applying the Kerberos protocol and provide credibility to access the services provided by hybrid

cloud computing. Two scenarios were implemented: the first without legitimate access and the second without illegal access. The simulation time was one hour. Figure 9 shows the average (in Traffic Sent (packets/sec)) and average (in Traffic Received (packets/Sec.)). With time, the network reaches stability and there are no problems with the amount of Traffic Sent and Traffic Received. Figure 10 shows the amount of delay that occurs when there is a hacker that is less than the absence of a hacker because the Kerberos protocol separates the illegal node from the network and consequently the network delay time is less. Figure 11 shows a very important measure, which is response time. It means network latency, also known as "network latency," is the amount of time required for a packet to travel across a network path from a transmitter to a receiving host. Response time is an important problem in cloud computing applications for remote control as the user sends a query and waits for the system to respond. We note that the response time in the usual network without a hacker is less than the system response time when there is a hacker and this confirms the ability and efficiency of the Kerberos protocol to separate the illegal node from the network.

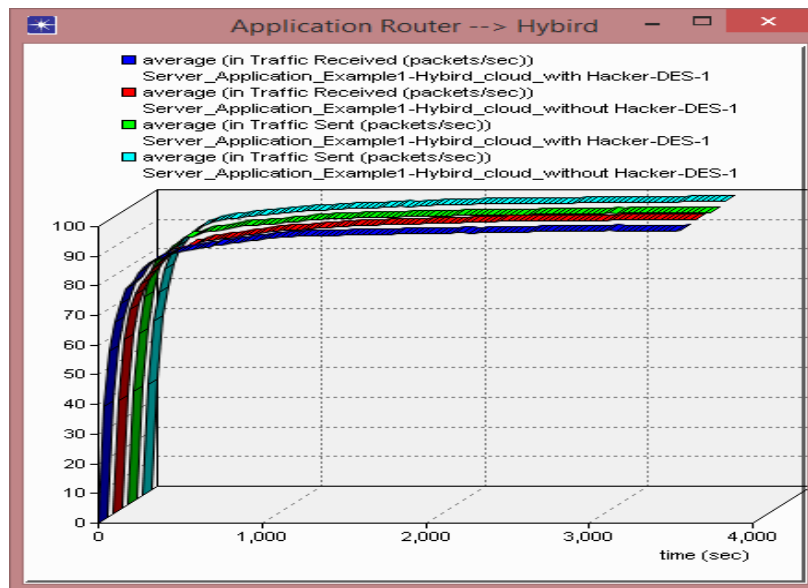


Figure 9: Traffic Sent/Received (packets/sec.)

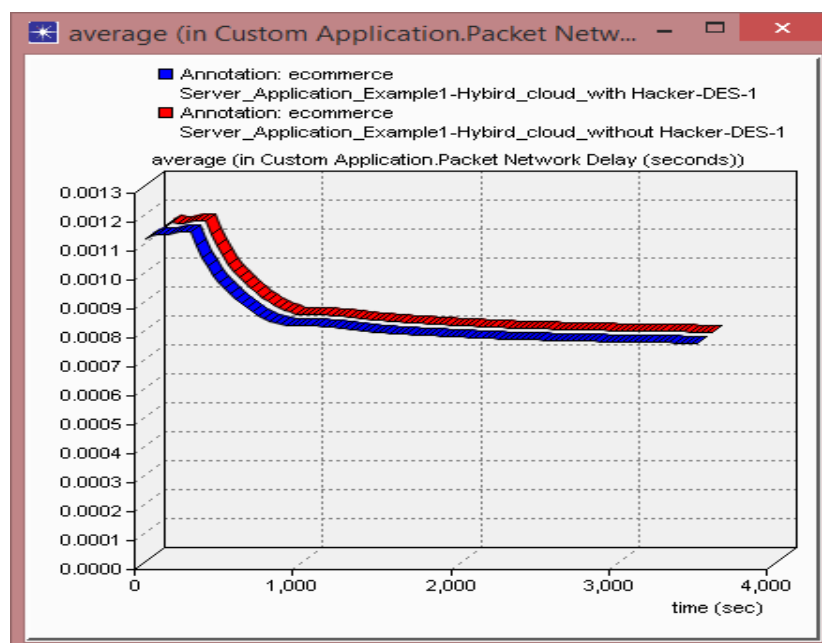


Figure 10: Application Packet Network Delay(Sec.)

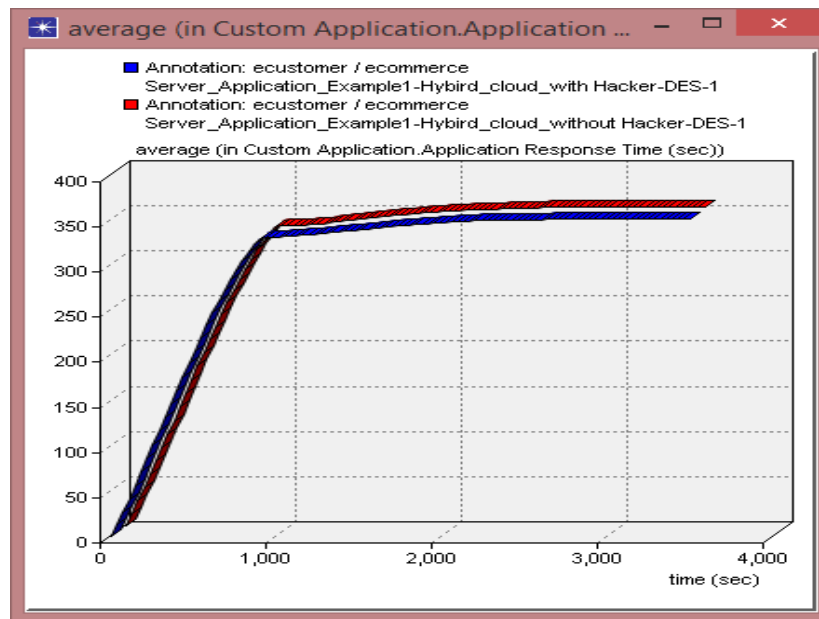


Figure 11: Application Response Time (sec)

## 5. CONCLUSIONS

Through this study, it was concluded that using Kerberos authentication in a hybrid cloud computing environment is very important. Although there is a lot of research in this area. However, this protocol has been extensively tested in a complex environment such as a hybrid cloud computing environment. However, the response of this model was better in preventing illegal access, which may contribute in the future to protecting network resources and increasing the confidence of users in the hybrid cloud. Also, the results achieved were satisfactory, especially concerning both network stability, delay time, and response time.

## References

- [1] Y. Mansouri, V. Prokhorenko, and M. A. Babar, "An Automated Implementation of Hybrid Cloud for Performance Evaluation of Distributed Databases", Preprint submitted to Journal of Network and Computer Applications, Vol. 167, 2020.
- [2] B. Saraladevia, N. Pazhanirajaa, P. Victor Paula, M.S. Saleem Bashab, P. Dhavachelvanc. "Big Data and Hadoop-A Study in Security Perspective," 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Procedia Computer Science 50, pp. 596 – 601, 2015.
- [3] K. E. Ali, Sh. A. Mazen, E.E. Hassanein, "A proposed hybrid model for adopting cloud computing in e-government", Future Computing and Informatics Journal, Vol. 3, 2018.
- [4] R. I. Tinini, Matias R. P. Santos, G. B. Figueiredo, and D. M. Batista, "5GPpy: A SimPy-based simulator for performance evaluations in 5G hybrid Cloud-Fog RAN architectures," Simulation Modelling Practice and Theory, Vol. 101, 102030, pp.1-23, May 2020.
- [5] H. Song, "Testing and Evaluation System for Cloud Computing Information Security Products", Procedia Computer Science, ScienceDirect, Vol. 166, pp. 84–87, 2020.
- [6] A. J. Choudhury, M. Sain, H. Jae-Lee, H. Lim, and P. Kumar, "A Strong User Authentication Framework for Cloud Computing," In the Proceedings of the 2011 IEEE Asia-Pacific Services Computing Conference, Jeju, Jeju Island K20orea (South), pp. 110-115, 2011.
- [7] A. H. Emam, "Additional Authentication and Authorisation using Registered Email-ID for Cloud Computing," International Journal of Soft Computing and Engineering, Vol. 3, No. 2, pp. 110-113, 2013.
- [8] R. K. Banyal, P. Jain, and V. K. Jain, "Multifactor Authentication Framework for Cloud Computing," In the Proceedings of the Fifth International Conference on Computational Intelligence, Modelling and Simulation, IEEE, pp.105-110, 2013.
- [9] S. H. Khan, and M. A. Akbar, "Multi-Factor Authentication on Cloud," In the Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, 1-7, 2015.

- [10] A. M. Abdul, S. Jena, and M. Balraju, "Dual Factor Authentication To Procure Cloud Services, " In the Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing(PDGC), IEEE, 2016.
- [11] J. Zhao, Y. Lin, , J. Li, and C. Lei, "Kerberos based authentication for multiple clouds, " Journal of Computational Information Systems, Vol. 11, No. 11, pp. 4065-4077, 2015.
- [12] B. Patel, and P. Dubey, "Kerberos Authentication Protocol Modeling Using Nusmv Model," International Journal of Management, IT and Engineering, Vol. 2, No. 5, pp. 257-269, 2012.
- [13] D. Rogers. "Orthus v2 Authentication Protocol Enhancement, and Supporting Enterprise Architecture," The 2nd International Workshop on Privacy and Security in HealthCare (PSCare15), Procedia Computer Science 63, pp. 581 – 588, 2015.
- [14] . V. Impe, "Kerberos Made Easy," Retrieved on May 26, 2017 in the internet, security, [Online]. Available: <https://www.vanimpe.eu/2017/05/26/kerberos-made-easy./> ,2017.
- [15] D. Cooper; S. Santesson; S. Farrell; S. Boeyen; R. Housley; and W. Polk Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC 5280. May 2008. [Online]. Available: <https://tools.ietf.org/html/rfc5280> (accessed on 28 July 2020)