Network as a Service Model in Cloud Authentication by HMAC Algorithm

Bommala Harikrishna
Ph.D Full Time Research Scholar, Research ID: M-7489-2017
Department of Computer Science and Engineering, YSR Engineering College of Yogi Vemana University Proddatur
Kadapa, Andhra Pradesh, India- 516360
Email: haribommala@gmail.com
Dr. S. Kiran
Assistant Professor, Department of Computer Science and Engineering,
YSR Engineering College of Yogi Vemana University Proddatur
Kadapa, Andhra Pradesh, India- 516360
Email: rkirans125 @gmail.com
K. Mani Deep
Assistant Professor, Department of Computer Science and Engineering,
Bapatla Engineering College, Bapatla, AP, India- 522101
Email: manideep.karumanchi@gmail.com

ABSTRACT

Resource pooling on internet-based accessing on use as pay environmental technology and ruled in IT field is the cloud. Present, in every organization has trusted the web, however, the information must flow but not hold the data. Therefore, all customers have to use the cloud. While the cloud progressing info by securing-protocols. Third party observing and certain circumstances directly stale in flow and kept of packets in the virtual private cloud. Global security statistics in the year 2017, hacking sensitive information in cloud approximately maybe 75.35%, and the world security analyzer said this calculation maybe reached to 100%. For this cause, this proposed research work concentrates on Authentication-Message-Digest-Key with authentication in routing the Network as a Service of packets in OSPF (Open Shortest Path First) implementing Cloud with GNS3 has tested them to securing from attackers.

Keywords – **Authentication, Attacks, Cloud, Cryptography, Protocols, Router, Security**

Date of Submission: March 05, 2018 Date of Acceptance: March 23, 2018

I. INTRODUCTION

In the year 2009, IT commercial organizations announced that Cloud was the second place. The role in IT industries has to sustain data centers and cloud profoundly infrastructures for rendering like Digital Ocean, Amazon Web Service, Yard, Wireshark, and Microsoft Azure. Through internet to service utility computing to software service by data centers to obtaining the internet to access the application services. The Hardware as a Service (HaaS) infers a virtual provider with respect to registering assets like OSs, organizing, Infrastructure, virtualization innovation, and [2] capacity administrations. Noblest IaaS benefit provider by AWS. This association can be overviewed as the fundamental system for another two associations, for example, PaaS and SaaS. In Application as a Service (SaaS), buyers or customers contract programming encouraged by the expert organization. It comes to pass web based on installment [6]. If providers started using the virtual private cloud network (VPCN) for communication [3]. IT infrastructures such as the spread of wired/wireless broadcasting network moreover [6] the variety of toolsets, high speed, and the scope of free software and so on. The customer has faced with problems of authentication in a cloud model of service in Authentication as a Service [9].

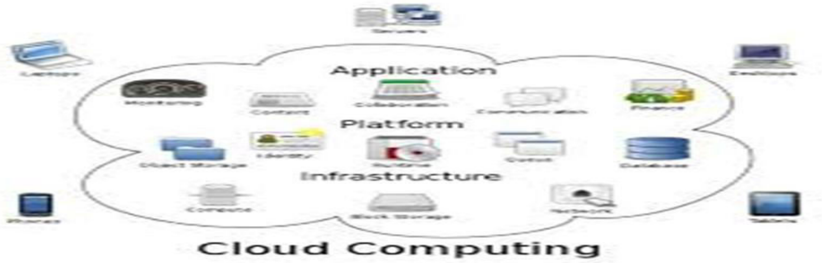


Fig 1: Cloud Computing

There are bunches of components and devices are recently connected yet, we are confronting numerous sorts of issues in the web world. Into a neighborhood machine IPV4 arrange designed to get to the cloud and switch setup to keep up the username and secret word for verification to center around inquire about for validation for packets [10].

The analysis shows that to be able to offer secured neighborhood communique then plaintext passwords and keyed MD5 authentication is wished. Plaintext passwords are prone to eavesdropping, at the [11] same time as keyed MD5 authentication can successfully protect network protocol exchanges.

II. AUTHENTICATION AS A SERVICE IN CLOUD FOR USER

Service model cloud for new users has accessed any resource pooling fully depend on the authentication. In Generally service provider mention the registration for privacy info after, providers give customer ID for identification, and method for authentication for customer authentication finished for registration done. When service utilizing in a client, that time asking the authentication and ID of the client. In AaaS maintained by the strong authentication in some possibility way of authentication process has been hacked by the hacker, that reason AaaS one part of PaaS maintain safe techniques like PSWD, MFA(Multi-Factor-Authentication), CAPKI (Crypto-Authentication-Public-Key-Infrastructure), SOO(Sign-On-One), OTP (One-Time-Password) and SBMM (Scan-Bar-Code-Mobile-Method).

2.1 weakness of Authentication as a Service in Cloud:

PSWD: PSWD nothing but a password, is a one of Authentication as a Service. Now a days electronics device has maintained the password for security. In some situations is a complication and immediately hold the renewal password for security reasons.

MFA: Multi-Factor-Authentication is a kind of Authentication as a Service. MFA was clubbed with different methods like password, iris, OTP, certificate and biometrics.

CAPKI: Crypto-Authentication-Public-Key-Infrastructure is among one of in an Authentications as a Service. Based on the certificate to authenticate third party, but don't share the safe info. The client way of to manage and inspect are not possible by the CAPKI.

SOO: Sign-On-One is a kind of Authentication as a Service. If authenticate any one application, then assertion access through the other side.

MTM: Mobile Trusted Module is proposed by TCG which Samsung, Nokia, Ericson etc., which based on SIM with authentication for Smartphone.

2.2 Dangerous Attack for Authentication:

An inflow of data packet through the internet from source to destination in that time the third party attacking the packets during the network.

Active Attacks: In computing security, an active attack is characterized by the attacker attempting to interrupt into the gadget. At some stage in an energetic attack, the intruder will introduce statistics into the device, in addition to doubtlessly trade records in the gadget.

Dos Attacks: The attacks paintings via asking for such a lot of resources from a server that the server cannot reply to valid requests. As Dos assault that originates from a single tool.

Distributed Attacks: Inside the beyond, recoveries had been limited to the processing energy of only one machine. It has to calculate the energy of the machine by using DNA and around the globe to decrypt passwords.

Close in Attacks: One kind of close in assaults is social engineering attacks wherein the attacker uses information of the other persons and attempts to take advantage of them with the aid of sending an e-mail or telephone and tries to acquire confidential facts [18] about their bank bills and so forth. In close-in-assaults, the attacker tries to gain records from bodily entities or community components by way of getting bodily near.

DDoS Attacks: Now and again protection specialists can purpose command and manage laptop structures inner a botnet, disrupting operations. But, figuring out command and manipulate devices takes time. Even postulate those are recognized then eliminated, partial botnets are designed in imitation of slip off regarding a failed command [19].

III. ARCHITECTURE FOR CLOUD WITH GNS3

GNS3 (Graphical Network Simulator), the remarkable network simulation software permits you after be part of then cooperate virtual network topologies after actual networks [12]. This exquisite feature of GNS3 brings CCNA, CCNP then CCIE labs [4] with absence hardware! Individually, I was plenty severe then got here in imitation of understand that opportunity inside GNS3. You may even associate in [1] accordance with digital machines past VMware then virtual container to this robust section on software. To tell the Idle computer charge into GNS3 going for walks above the windows platform [5]. Linking a virtual topology into GNS3 to actual devices is at all exciting or sturdy then as makes enterprise network setup nicely.

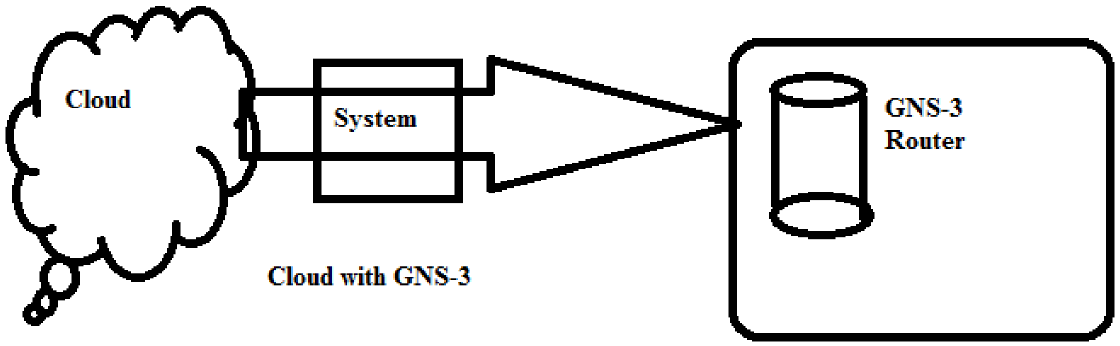


Fig 2: Cloud with GNS3

In cloud, internet network based on applications so, networking is a configured with an IP address in IPV4 in local system [16]. That IP address configure within Cloud. In business market simulation routers have so many, but GNS3 provide routers like 1720,1710,2660,3675 and 7200 and so on. But unique ideal number generally generates for each every time in a router for every user [13]. On GNS3 connection with cloud based upon IP address with the port number and configure router. On each and every router has a open by using putty software.

3.1 Message Digest Key Authentication Routing Authentication:

In attacker's security reason for using MD-5 Cryptography-Key-Algorithms.MD-5 algorithm partition into the modules, each module block size of input 512 bits. At the end of the last block of module join the 64 bits. The genuine input date, length recording purpose used the 64 bits. If the last block is less than 512 bits, some extra bits are 'padded' to the end. Each block contains sixteen words of 32 bits each [11]. It is the representation as A0, A1, A2, A3… A15.

The buffer is part of MD-5, creates 4 words and each holds 32 bits long. The listed below

i.	ONE:	01 23 45 67	(A)
ii.	TWO:	89 ab CD ef	(B)
iii.	THREE:	fe dc ba 98	(C)
iv.	FOUR:	76 54 32 10	(D)

Trigonometry Sin function is calculate to enhancement on future utilizing so maintain the MD-5 table T.

Tj = abs(sin(j + 1)) * 232.

We consider 'T' has 64 elements.

Element number j is indicated as Tj.

AUXILIARY OF MESSAGE DIGEST KEY AUTHENTICATION ROUTING AUTHENTICATION:

It's used to logical operator ∧, ∨, 7, and, xor to the input data. Each function takes 3 32-bits and gives the output one 32-bit.

P(L,M,N) = (L and M) or (not(L) and N) →(F)

Q(L,M,N) = (L and N) or (M and not(N)) →(G)

R(L,M,N) = L xor M xor N →(H)

S(L,M,N) = M xor (L or not(N))→(I)

MD-5 have four rounds, each participate 16 operations below is one operation.

There are four rounds, each involves 16 basic operations

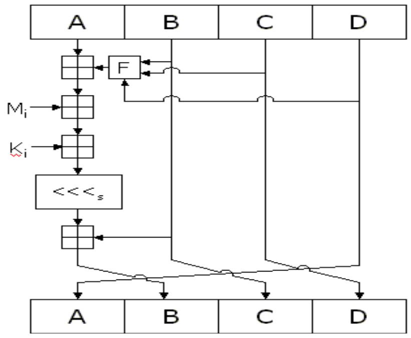


Fig 3: Message Digest Key Authentication Routing Authentication algorithm for single operation.

Message Digest Hash function algorithm generates a hash value to observing the OSPF routing packets and authenticate a password. The packets are transmitted within hash value attached to take a KEY-ID and non-decreasing the linear number [7]. The Receiver should know the same password of the sender, computes its own hash value. Third-party modifies the message without permission of the sender but, don't worry about the original message because of the hash value of should be same sender and receiver [15]. Multiple passwords allow the router so each password contains unique key-id. So all passwords are collaborating or migrate easy and secure is more. However, they do need to be the same between neighbors. The Authentication type area, that's configurable on a router per-interface foundation, [14] identifies the authentication set of rules. MD5 authentication at the sending router.

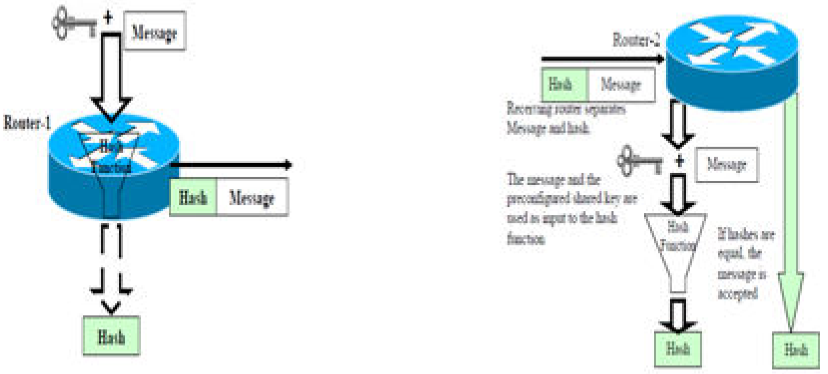

Message
Router-1
Hash Function
Hash
Message
Hash
Router-2
Hash
Message
Receiving router separates Message and hash.
Message
The message and the preconfigured shared key are used as input to the hash function
Hash Function
If hashes are equal, the message is accepted
Hash
Hash

Fig 4: Message Digest Key Authentication Sending and Receiver Router C3645.

IV. GNS3 WITH CLOUD IMPLEMENTATION BY PROPOSED ALGORITHM AND EXPERIMENT

In this research and analyzing work, by the outside network to internal network request from connection to destination system in flood attackers. Before starting the implementation of Cloud with GNS3 authentication by using the md5 cryptography algorithm, has required the open source latest version GNS3 [8] with browser the router and Loopback Network Adapter with IP address assigning with virtual cloud configure with nodes in NIO Ethernet. In ISO, Cisco C3745 dynamics GT96100-FE router images are downloaded in GNS3 open source website. In router configure with Ethernet, Fast Ethernet and Serial. Each router has a unique idle number [4]. In Gns3 with the cloud router has contains Telnet 127.0.0.1: port numbers 5000, 5001, 5003 and so on and inner configure python 3.6.3 version.

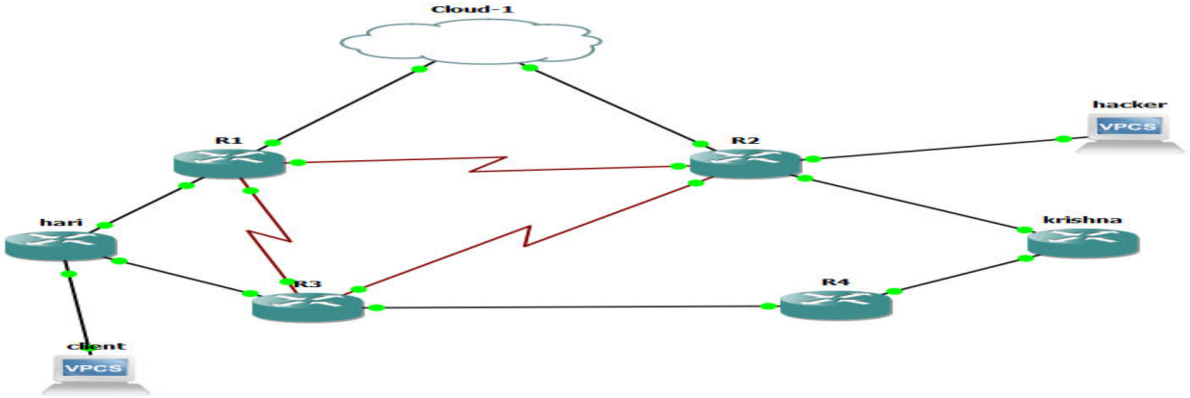
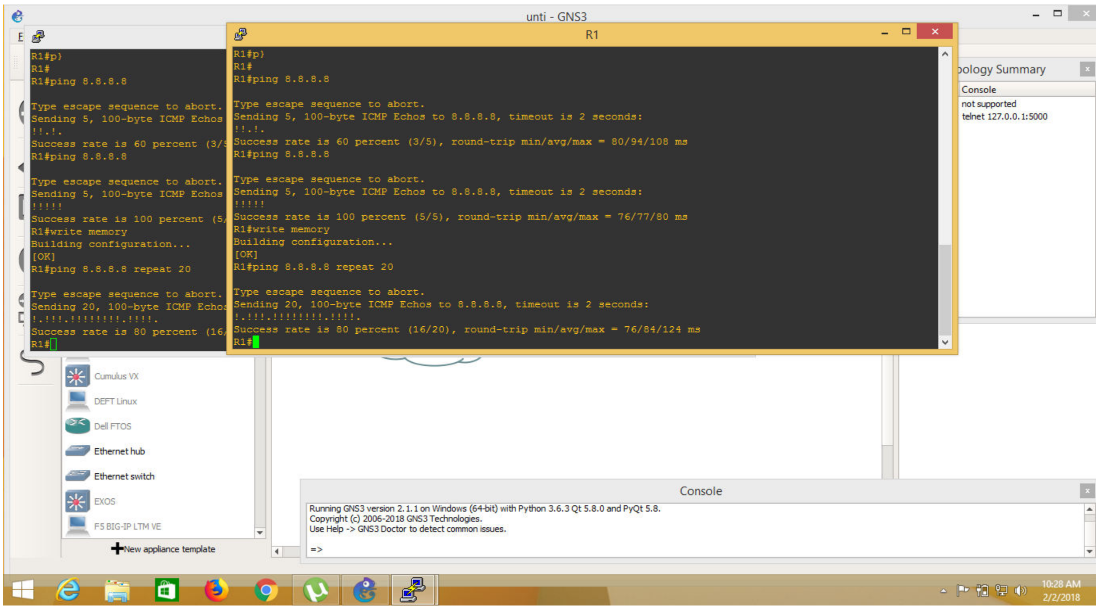


Fig 5: Configure Cloud with Network as a Service.

After completion of configuration check the cloud is connected or not, then open which router is connect the cloud by using putty.

```
R1# ping 8.8.8.8 repeat 20
Sending 5, 100-byte echo to 8.8.8.8 time out to reach 2 seconds.
!,!!!,!!!!!!!!,!!!,!
```

The Success rate is 80 percent (16/20), round-trip min/avg/max = 74/81/124 ms.



```
R1#p}
R1#
R1#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!.!.
Success rate is 60 percent (3/5), round-trip min/avg/max = 80/94/108 ms
R1#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/77/80 ms
R1#write memory
Building configuration...
[OK]
R1#ping 8.8.8.8 repeat 20

Type escape sequence to abort.
Sending 20, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!.!!!.!!!!!!!!.!!!!.
Success rate is 80 percent (16/20), round-trip min/avg/max = 76/84/124 ms
R1#
```

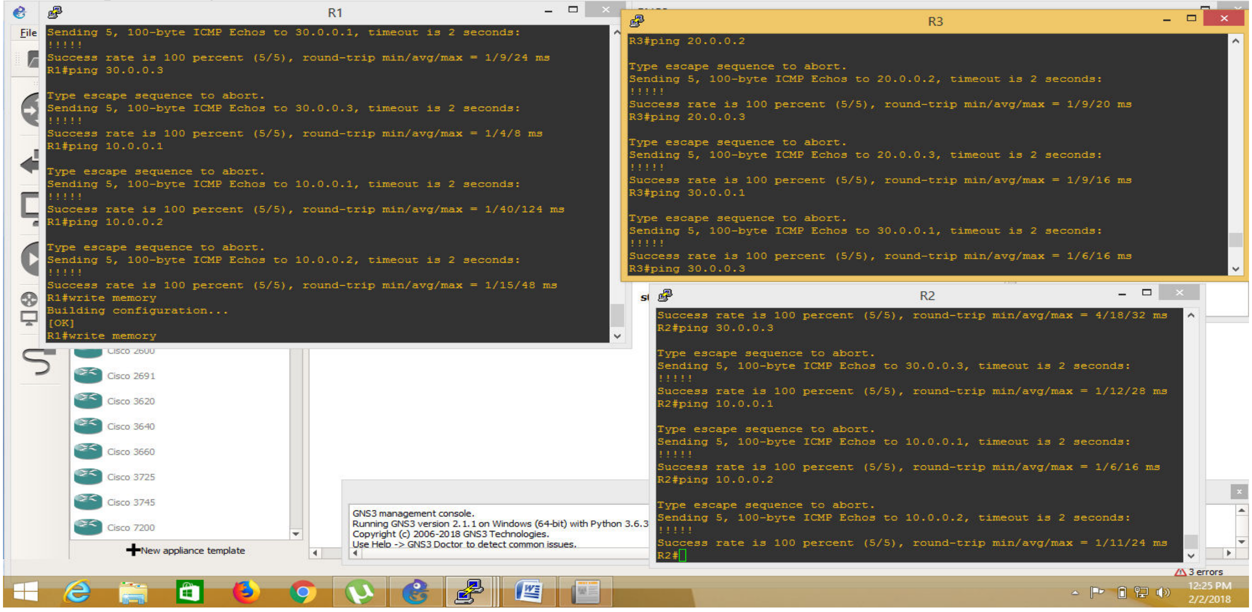
```
Running GNS3 version 2.1.1 on Windows (64-bit) with Python 3.6.3 Qt 5.8.0 and PyQt 5.8.0.
Copyright (c) 2006-2018 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.
=>
```

Fig 6: Cloud connection by Putty with Router.

4.1 Steps for Router Configuration:

Step 1: open all routers using putty.
Step 2: enable router configuration.
Step 3: configuration terminal in router.
Step 4: By using Ethernet, Fast Ethernet, Serial and ATM any one configure the interface.
Step 5: set the IP address and subnet mask.
Step 6: save the all configuration.

Above steps configuration to all the routers, after ping individual IP address. Calculate the individual min/ avg /max.



```
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/24 ms
R1#ping 30.0.0.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/8 ms
R1#ping 10.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/40/124 ms
R1#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/15/48 ms
R1#write memory
Building configuration...
[OK]
R1#write memory
```

```
R3#ping 20.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/20 ms
R3#ping 20.0.0.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/16 ms
R3#ping 30.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/16 ms
R3#ping 30.0.0.3
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/18/32 ms
R2#ping 30.0.0.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/12/28 ms
R2#ping 10.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/16 ms
R2#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/11/24 ms
R2#
```

```
GNS3 management console.
Running GNS3 version 2.1.1 on Windows (64-bit) with Python 3.6.3
Copyright (c) 2006-2018 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.
```

Fig 7: Connecting all Routers with IP address Success Rate.

4.2 Proposed Algorithm Implementation for Ospf Authentication-Key with Message Digest Algorithm Cloud:

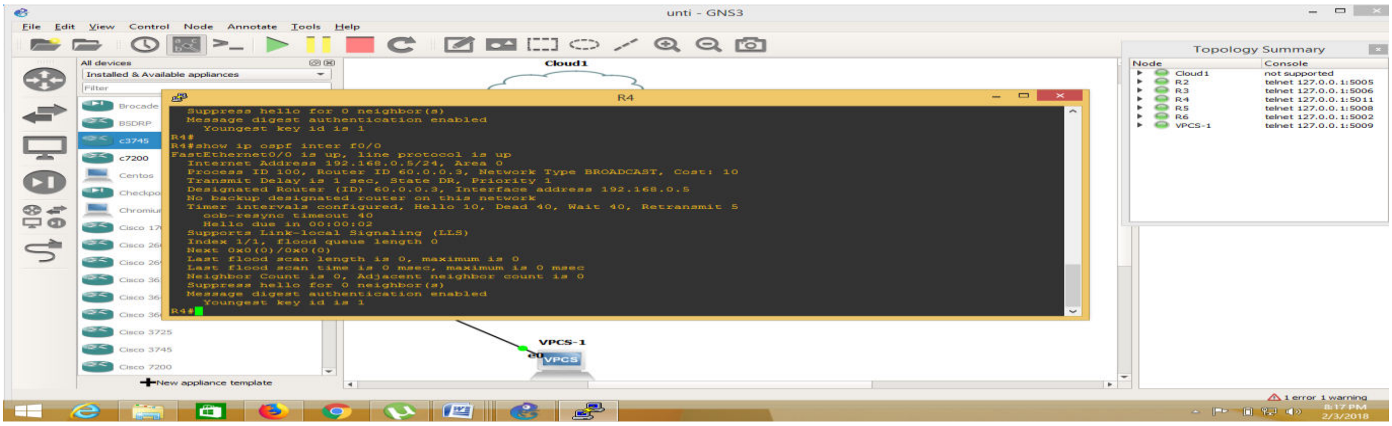
Algorithm: pre-requests are VPCS =VP{VP1,VP2,. .. VPn}, Routers R := {series C3745 routers, r1, r2, r3 …rn} and Cloud C:= {Cloud-1, Cloud-2….. Cloud-n}. below steps from 1 to 13.

```
1.  Routing = {OSPF}, O={Non-Securing and MD5 Securing}.
2.  Set D = Routing x O where
3.  Loop { for all d belongs to D} :
4.  Loop { r m belongs to R, where i belongs to {1,2,3,…m}
5.  Set up and config d on ri
6.  Start server with IPx port,
7.  Loop {for all VPCSj = VP, where j = 1,2,…n}
8.  Establish connection to Cloud at IPx port.
9.  Set up all R x Routing { r1,r2… rn with OSPF}
10. Set up D with Router
11. Loop { r m belongs to R, where i belongs to {1,2,3,…m} and inner dm belongs to D is {OSPF x O}
12. Loop do simulation.
13. Stop
```

Commands to Establishment for Ospf Authentication Message Digest Algorithm:

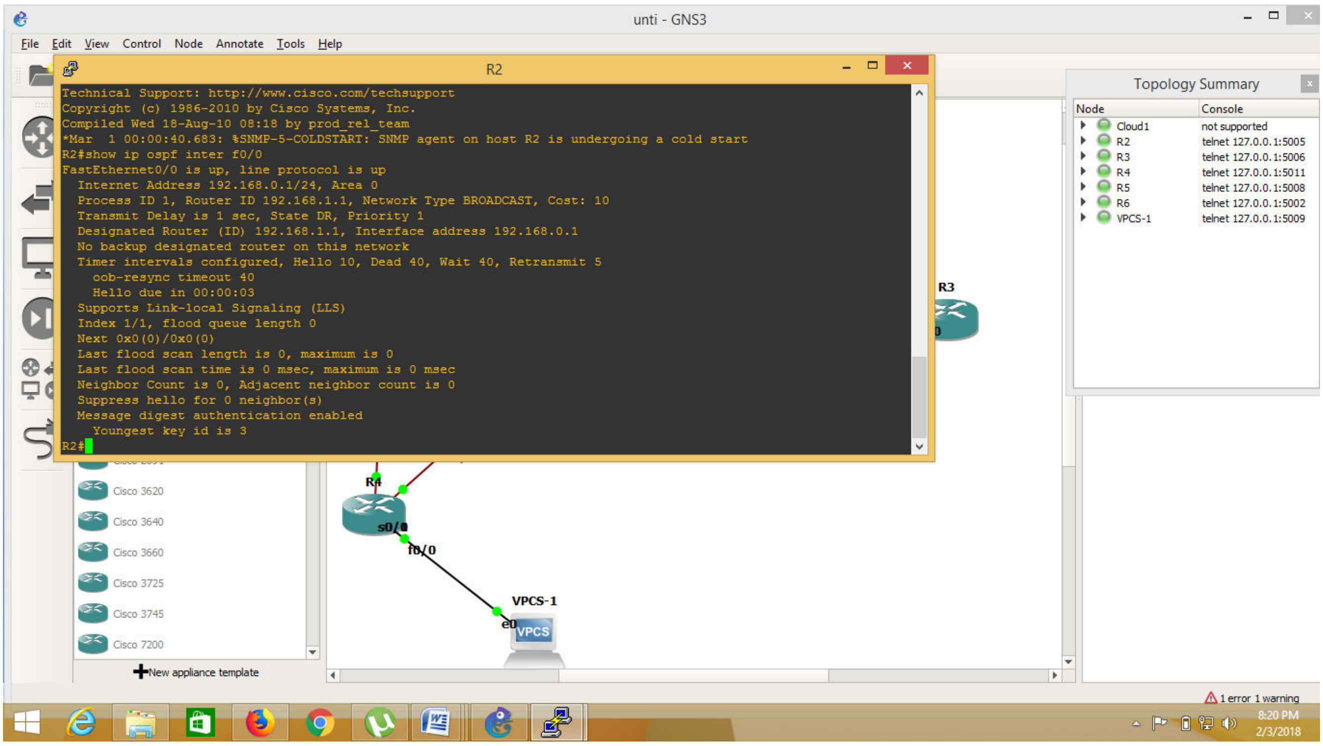
1. Enables the routers(the initial router has not secure configure the authentication)
2. Configuration with terminal
3. Interface with Ethernet.
4. Configuration each router IP address with subnet mask.
5. Set the router OSPF version.
6. Set the network with sub network mask.
7. Set and configure the OSPF with MD5 with key { 1 to 255} and OSPF password { max 16 characters}
8. Setting and configuration, authentication with MD5.
9. Verification show the IP OSPF neighbor.
10. Show the OSPF interface
11. Finally, debug the IP OSPF packets.

Below figure 8 shows that Router 4 is authenticated by MD-5 Cryptography Algorithm. Its have youngest key id is 1.



```
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1
R4#
R4#show ip ospf inter f0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 192.168.0.5/24, Area 0
  Process ID 100, Router ID 60.0.0.3, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 60.0.0.3, Interface address 192.168.0.5
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1
R4#
```

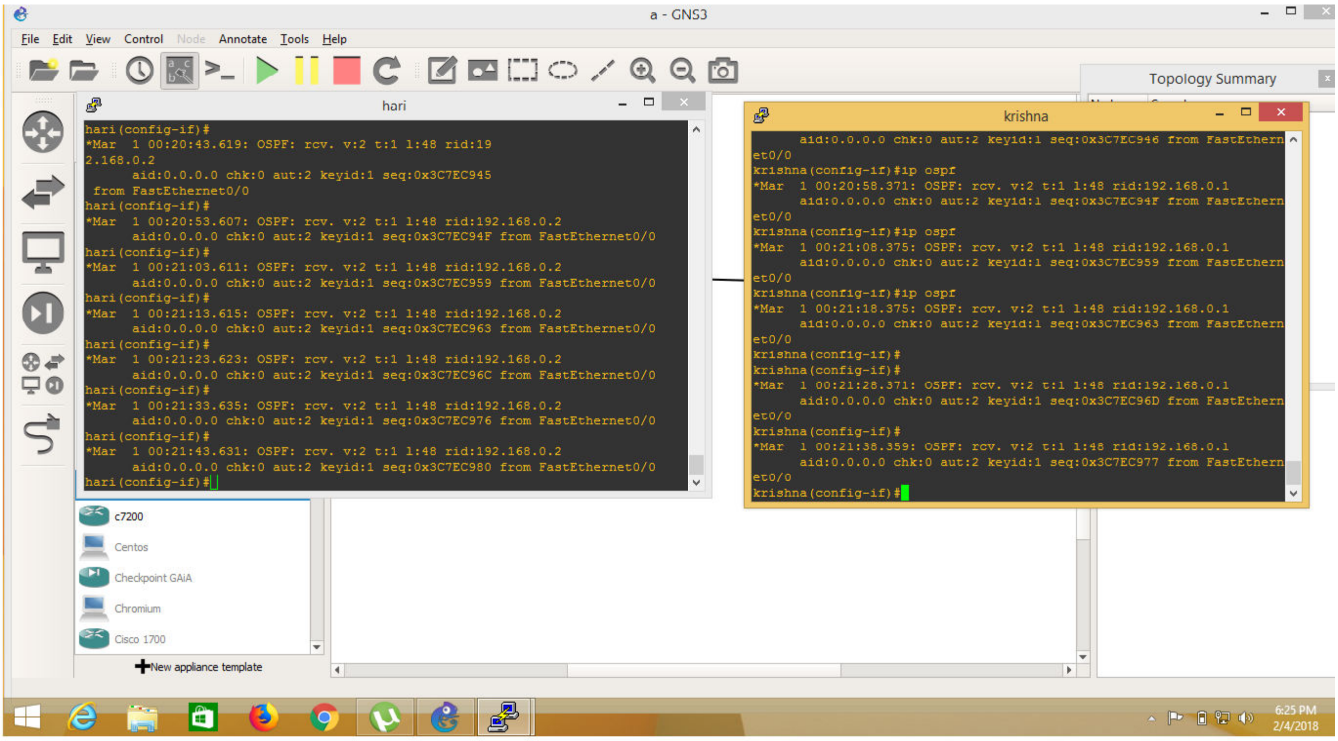
Fig 8: Message Digest Authentication enables in Router 4:



```
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Wed 18-Aug-10 08:18 by prod_rel_team
*Mar  1 00:00:40.683: %SNMP-5-COLDSTART: SNMP agent on host R2 is undergoing a cold start
R2#show ip ospf inter f0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 192.168.0.1/24, Area 0
  Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.1.1, Interface address 192.168.0.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 3
R2#
```

Fig 9: Message Digest Authentication enables router R2.

Above figure 9 shows that Router 2 is authenticated by MD-5 Cryptography Algorithm. In R2 router has Key id is 3.



```
hari(config-if)#
*Mar  1 00:20:43.619: OSPF: rcv. v:2 t:1 l:48 rid:19
2.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC945
 from FastEthernet0/0
hari(config-if)#
*Mar  1 00:20:53.607: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC94F from FastEthernet0/0
hari(config-if)#
*Mar  1 00:21:03.611: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC959 from FastEthernet0/0
hari(config-if)#
*Mar  1 00:21:13.615: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC963 from FastEthernet0/0
hari(config-if)#
*Mar  1 00:21:23.623: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC96C from FastEthernet0/0
hari(config-if)#
*Mar  1 00:21:33.635: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC976 from FastEthernet0/0
hari(config-if)#
*Mar  1 00:21:43.631: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.2
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC980 from FastEthernet0/0
hari(config-if)#
```

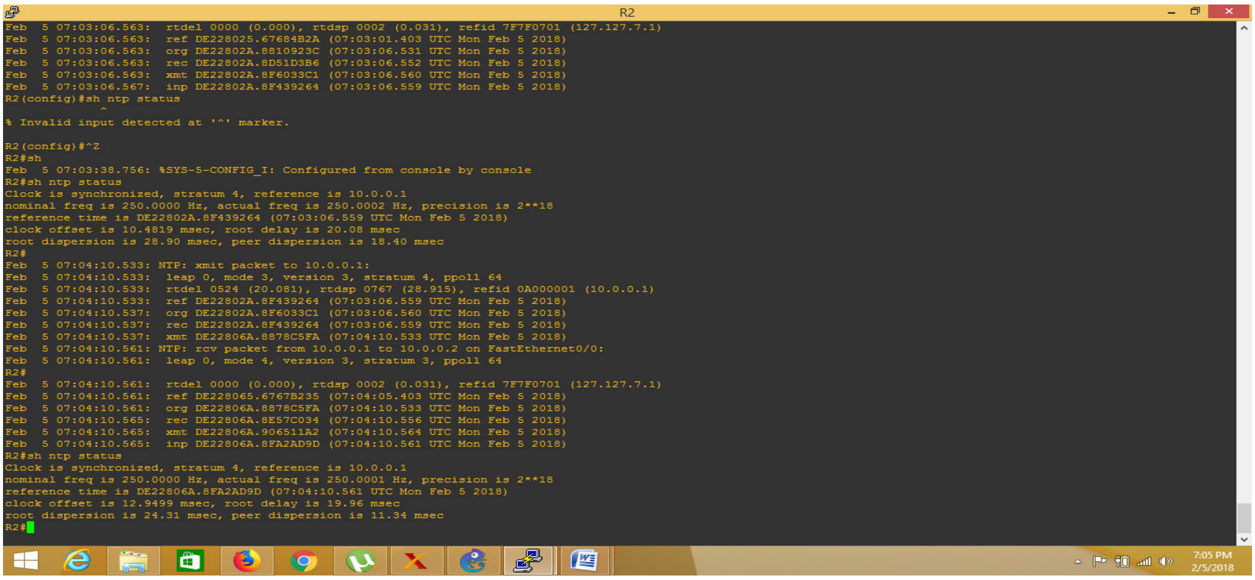
```
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC946 from FastEthern
et0/0
krishna(config-if)#ip ospf
*Mar  1 00:20:58.371: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.1
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC94F from FastEthern
et0/0
krishna(config-if)#ip ospf
*Mar  1 00:21:08.375: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.1
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC959 from FastEthern
et0/0
krishna(config-if)#ip ospf
*Mar  1 00:21:18.375: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.1
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC963 from FastEthern
et0/0
krishna(config-if)#
krishna(config-if)#
*Mar  1 00:21:28.371: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.1
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC96D from FastEthern
et0/0
krishna(config-if)#
*Mar  1 00:21:38.359: OSPF: rcv. v:2 t:1 l:48 rid:192.168.0.1
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EC977 from FastEthern
et0/0
krishna(config-if)#
```

Fig 10: Authenticate for IP address OSPF packets.

Above figure 10 shows the packet secure communication from source IP(192.168.0.1) to destination IP(192168.0.2). Here source router is Hari and destination router id Krishna.

In Hari router received rid: 192.168.0.2 aut : 2 keyid:1 seq : **0x3C7ECC80** from FastEthernet0/0.

In Krishna router received rid: 192.168.0.2 aut : 2 keyid:1 seq **: 0x3C7EC77** from FastEthernet0/0.

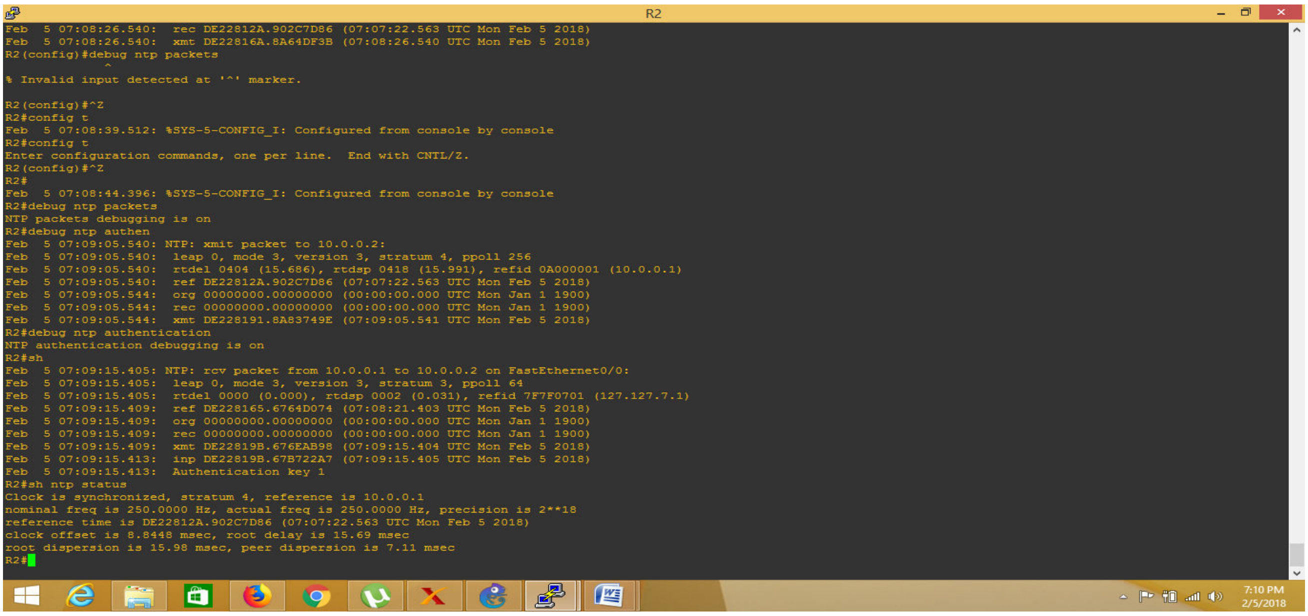


```
Feb  5 07:03:06.563:  rtdel 0000 (0.000), rtdsp 0002 (0.031), refid 7F7F0701 (127.127.7.1)
Feb  5 07:03:06.563:  ref DE228025.67684B2A (07:03:01.403 UTC Mon Feb 5 2018)
Feb  5 07:03:06.563:  org DE22802A.8810923C (07:03:06.531 UTC Mon Feb 5 2018)
Feb  5 07:03:06.563:  rec DE22802A.8D51D3B6 (07:03:06.552 UTC Mon Feb 5 2018)
Feb  5 07:03:06.563:  xmt DE22802A.8F6033C1 (07:03:06.560 UTC Mon Feb 5 2018)
Feb  5 07:03:06.567:  inp DE22802A.8F439264 (07:03:06.559 UTC Mon Feb 5 2018)
R2(config)#sh ntp status
             ^
% Invalid input detected at '^' marker.

R2(config)#^Z
R2#sh
Feb  5 07:03:38.756: %SYS-5-CONFIG_I: Configured from console by console
R2#sh ntp status
Clock is synchronized, stratum 4, reference is 10.0.0.1
nominal freq is 250.0000 Hz, actual freq is 250.0002 Hz, precision is 2**18
reference time is DE22802A.8F439264 (07:03:06.559 UTC Mon Feb 5 2018)
clock offset is 10.4819 msec, root delay is 20.08 msec
root dispersion is 28.90 msec, peer dispersion is 18.40 msec
R2#
Feb  5 07:04:10.533: NTP: xmit packet to 10.0.0.1:
Feb  5 07:04:10.533:  leap 0, mode 3, version 3, stratum 4, ppoll 64
Feb  5 07:04:10.533:  rtdel 0524 (20.081), rtdsp 0767 (28.915), refid 0A000001 (10.0.0.1)
Feb  5 07:04:10.533:  ref DE22802A.8F439264 (07:03:06.559 UTC Mon Feb 5 2018)
Feb  5 07:04:10.537:  org DE22802A.8F6033C1 (07:03:06.560 UTC Mon Feb 5 2018)
Feb  5 07:04:10.537:  rec DE22802A.8F439264 (07:03:06.559 UTC Mon Feb 5 2018)
Feb  5 07:04:10.537:  xmt DE22806A.8878C5FA (07:04:10.533 UTC Mon Feb 5 2018)
Feb  5 07:04:10.561: NTP: rcv packet from 10.0.0.1 to 10.0.0.2 on FastEthernet0/0:
Feb  5 07:04:10.561:  leap 0, mode 4, version 3, stratum 3, ppoll 64
R2#
Feb  5 07:04:10.561:  rtdel 0000 (0.000), rtdsp 0002 (0.031), refid 7F7F0701 (127.127.7.1)
Feb  5 07:04:10.561:  ref DE228065.6767B235 (07:04:05.403 UTC Mon Feb 5 2018)
Feb  5 07:04:10.561:  org DE22806A.8878C5FA (07:04:10.533 UTC Mon Feb 5 2018)
Feb  5 07:04:10.565:  rec DE22806A.8E57C034 (07:04:10.556 UTC Mon Feb 5 2018)
Feb  5 07:04:10.565:  xmt DE22806A.906511A2 (07:04:10.564 UTC Mon Feb 5 2018)
Feb  5 07:04:10.565:  inp DE22806A.8FA2AD9D (07:04:10.561 UTC Mon Feb 5 2018)
R2#sh ntp status
Clock is synchronized, stratum 4, reference is 10.0.0.1
nominal freq is 250.0000 Hz, actual freq is 250.0001 Hz, precision is 2**18
reference time is DE22806A.8FA2AD9D (07:04:10.561 UTC Mon Feb 5 2018)
clock offset is 12.9499 msec, root delay is 19.96 msec
root dispersion is 24.31 msec, peer dispersion is 11.34 msec
R2#
```

Fig 11: Clock offset, root delay, root dispersion and peer dispersion without uthentication.

In above figure : 11 shows that, the designing authentication as a service model packets transmitted by using the OSPF routing protocol, packets reached the destination without using the authentication algorithm in R2 reference is 10.0.0.1.Now by using Network Time Protocol to analyzing the Clock offset, root delay, root dispersion and peer dispersion of the time calculated.



```
Feb  5 07:08:26.540:  rec DE22812A.902C7D86 (07:07:22.563 UTC Mon Feb 5 2018)
Feb  5 07:08:26.540:  xmt DE22816A.8A64DF3B (07:08:26.540 UTC Mon Feb 5 2018)
R2(config)#debug ntp packets
             ^
% Invalid input detected at '^' marker.

R2(config)#^Z
R2#config t
Feb  5 07:08:39.512: %SYS-5-CONFIG_I: Configured from console by console
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#^Z
R2#
Feb  5 07:08:44.396: %SYS-5-CONFIG_I: Configured from console by console
R2#debug ntp packets
NTP packets debugging is on
R2#debug ntp authen
Feb  5 07:09:05.540: NTP: xmit packet to 10.0.0.2:
Feb  5 07:09:05.540:  leap 0, mode 3, version 3, stratum 4, ppoll 256
Feb  5 07:09:05.540:  rtdel 0404 (15.686), rtdsp 0418 (15.991), refid 0A000001 (10.0.0.1)
Feb  5 07:09:05.540:  ref DE22812A.902C7D86 (07:07:22.563 UTC Mon Feb 5 2018)
Feb  5 07:09:05.544:  org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
Feb  5 07:09:05.544:  rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
Feb  5 07:09:05.544:  xmt DE228191.8A83749E (07:09:05.541 UTC Mon Feb 5 2018)
R2#debug ntp authentication
NTP authentication debugging is on
R2#sh
Feb  5 07:09:15.405: NTP: rcv packet from 10.0.0.1 to 10.0.0.2 on FastEthernet0/0:
Feb  5 07:09:15.405:  leap 0, mode 3, version 3, stratum 3, ppoll 64
Feb  5 07:09:15.405:  rtdel 0000 (0.000), rtdsp 0002 (0.031), refid 7F7F0701 (127.127.7.1)
Feb  5 07:09:15.409:  ref DE228165.6764D074 (07:08:21.403 UTC Mon Feb 5 2018)
Feb  5 07:09:15.409:  org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
Feb  5 07:09:15.409:  rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
Feb  5 07:09:15.409:  xmt DE22819B.676EAB98 (07:09:15.404 UTC Mon Feb 5 2018)
Feb  5 07:09:15.413:  inp DE22819B.67B722A7 (07:09:15.405 UTC Mon Feb 5 2018)
Feb  5 07:09:15.413:  Authentication key 1
R2#sh ntp status
Clock is synchronized, stratum 4, reference is 10.0.0.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
reference time is DE22812A.902C7D86 (07:07:22.563 UTC Mon Feb 5 2018)
clock offset is 8.8448 msec, root delay is 15.69 msec
root dispersion is 15.98 msec, peer dispersion is 7.11 msec
R2#
```

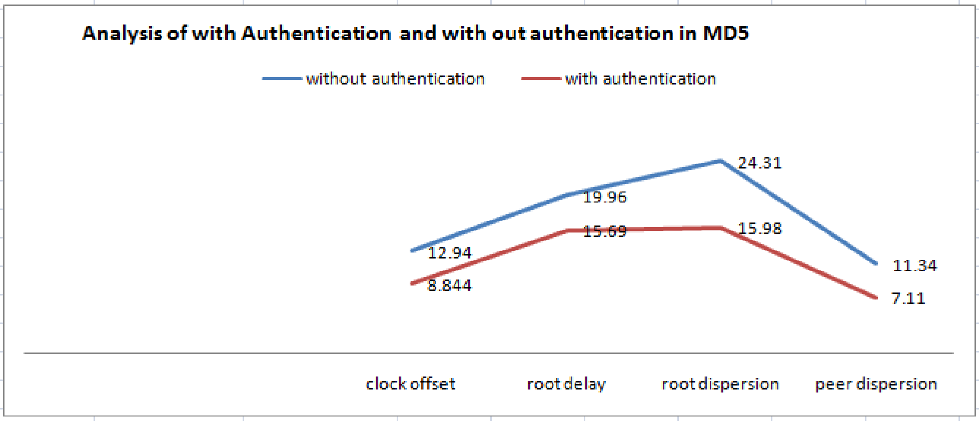
Fig 12: Clock offset, root delay, root dispersion and peer dispersion with Authentication.

In above figure: 12 shows that, the designing authentication as a service model packets transmitted by using the OSPF routing protocol, packets reached the destination by using authentication algorithm. In R2, receive packet from 10.0.0.1 to 10.0.0.2 on FastEthernet0/0 by using the Network Time Protocol to analyzing the Clock offset, root delay, root dispersion and peer dispersion of the time calculated.

The below table: 1 mention values are calculated by using NTP in OSPF routing in the Cloud.

Time	Clock offset	Root delay	Root dispersion	Peer dispersion
without authentication	12.94	19.96	24.31	11.34
with authentication	8.844	15.69	15.98	7.11

Table 1: The result offset, root delay, root dispersion, peer dispersion



Graph 1: Graph for with authentication and without authentication

The above graph shows that in without authentication, the root delay, time offset, root dispersion and peer dispersion is increasing. By using a proposed implementation cryptography algorithm the root delay, and root delay, root dispersion and peer dispersion ratio are reduced.

V. CONCLUSION

Security is the major aspect in the cloud computing. Every minute new attacks are generated in the concerned field, So, it requires the strongest security mechanism, to handle the all types of attacks. So, this paper proposes the strongest security mechanism to implement GNS3 simulator. GNS3 supports the authenticating the way of packets transmitted in the network within the cloud by using MD5 cryptographic algorithm along with Identity-Key for verification. The information is routed by using the OSPF protocol the service model of the Network as a Cloud. If choosing any path from the model, then MD5 had hid the packets and authenticated. The design model using the components of Cisco C3745 routers with interfacing serials, Fast-Ethernet with connecting putty terminal. Testing time comparison of time delay, root dispersion and peer dispersion, so gives best performance analysis in MD-5. GNS-3 is a graphical community simulator, which facilitates in, going for walks simulations at the consumer-designed models.

REFERENCES

[1] X. Wang and S. Zhang, Research about optimization of campus network security system, Procedia Eng., vol. 15, pp. 1802–1806, 2011.

[2] B.Harikrishna, S.Kiran, G.Murali and R.Pradeep Kumar Reddy, Security Issues In Service Model Of Cloud Computing Environment, Procedia Computer Science 87 (2016) 246 251, Science Direct.

[3] Cataldo Basile, Antonio Lioy, Analysis of Application Layer Filtering Policies With Application to HTTP, IEEE/ ACM Transactions on Networking, 1063-6692, 2013 IEEE.

[4] Free CCNA Tutorials. Study CCNA For Free!. Study-ccna.com. N.p., 2017. Web. 21 Mar. 2017.

[5] Cite A Website - Cite This For Me. Networkstraining.com. N.p., 2017. Web. 21 Mar. 2017.

[6] B.Harikrishna, N.Anusha, K.Manideep, Madhusudhanarao, Ch, Quarantine Stabilizing Multi-Keyword Rated Discover with Unfamiliar ID Transferover Encrypted Cloud Warning IJERCSE Vol 2, Issue 2, February 2015.

[7] Zili Shao, Chun Xue, Qingfeng Zhuge, Meikang Qiu, Bin Xiao, Edwin H.-M Sha, ―Security Protection and Checking for Embedded System Integration against Buffer Overflow attacks via Hardware/Software‖, IEEE Transactions on Computers, Vol. 55, NO. 4, April 2006.

[8] Source: https://docs.gns3.com/.
B. Harikrishna, Efficient Resource Allocation using Fair Scheduling in Cloud Based Systems, RITS ICA EM 2012.

[9] Chandra Wijaya 2011 *IEEE* 355-360.

[10] B. Harikrishna, S. Kiran, R. Pradeep Kumar Reddy, Protection on sensitive information in cloud Cryptography algorithms, IEEE digital Library 10.1109/CESYS.2016.7889894.

[11] Source: Computer weekly.com

[12] Jason C. Neumann The book of GNS3 Device Nodes, Live Switches, and the Internet 2015.

[13] Khalid Abu Al-Saud, Hatim Tahir, Moutaz Saleh and Mohammed Saleh 2010 *IAJIT* 380- 387.

[14] Check point FireWall-1, version 3.0 White paper June 1997 http://www.checkpoint.com/products/ whitepapers/wp30.pdf

[15] Wallace, Kevin. CCNP Routing And Switching ROUTE 300-101 Official Cert Guide. 1st ed. Indianapolis, IN: Pearson Education, 2015.

[16] Hon Sun Chiu, Kwan L. Yeung, and King-Shan Lui- J-CAR: An Efficient Joint Channel Assignment and Routing Protocol for IEEE 802.11-Based Multi-Channel Multi-Interface Mobile Ad Hoc Networks, 1706 IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 8 (4) , APRIL 2009.

[17] Anja Feldmann, Jennifer Rexford, and Ramon Caceres, ”Efficient Policies for Carrying Web Traffic Over Flow- Switched Networks” , IEEE/ ACM transactions on networking, vol. 6, no. 6, December 1998.

[18] Q.Zhao, Y. Mou, and S.H.. Qin, “The design of Security authentication system based on campus Network, “ Proc. – Int. Conf. Electr. Control Eng. ICECE 2010, pp. 3070-3073, 2010.