SSL VPN Performance Evaluation

By

Jihad Hisham Bashier Osman

A thesis submitted to University of Khartoum, of Engineering, Electrical and Electronic Engineering Department, in fulfillment of the requirements for the degree of

B.Sc. (HON)

In
Electrical and Electronic Engineering

Under Supervision of

Dr.Iman Abuel Maaly

July 2008

أهدا ء

الي كل من علمنا أن نغزل عطر الازهار
ان ننسج للفرح قميصاً بخيوط من نار
كي تهدأ خاطرة أولي كي يسكن نبض الاعصار
ليعود خريف مواجعنا
يندمل الحزن ..... الفقر..... الخوف..... الجهل..... وينهار

جهاد هشام بشير
يوليو 2008

Acknowledgement

I would like to express my sense of gratitude to my supervisor Dr. Iman Abual Maaly , for here continues advice systematic guidance , encouragement , and great support at all times . My deepest thank to my family for their significantly effort to provide a suitable environment to do my best.

I acknowledge my gratitude to my project partner Hafiz Eltayeb Elhag who makes large effort with to achieve this project

I acknowledge my gratitude to all stuff of IT department, Zain specially Eng. Nagi Yousif, to their advice and because they provided much appreciated helpful advice and information.

I acknowledge my gratitude to Eng. Nazar Abdalla Abraheem in IT stuff of MTN, who give us some experimental tests, under his supervision, to make last step in the project.

I'm deeply grateful to Eng. Omar Afifi in DataNet, how provide us with appreciated helpful advices during the preliminary research phase of this project.

I have greatly profited from hints, discussions and experiences generously lavished during the period of my university study from my friends and classmates.

Abstract

Virtual Private Networks (VPN) have become of increasing interest as technique to connect remote networks and users, this interesting is exist by economic perspective since the leased lines is not suitable solution economically to connect two far networks or support remote access to network resources and where the internet access is spread allover the world . However the VPN must secure enough since we use internet, which is open to all users ,make it easy to hack private data so we must using high security protocols , most recently VPN use IPsec tunneling protocol to give need security and privacy to intranet and extranet VPN ,but is it the best solution ? A Secure Socket Layer protocol is introduced to be new solution, but is SSL VPN degrades or increases the network performance?

This gives rise to the need for studies about SSL VPN to know the effects of the encryption of data and tunneling headers added to data be suitable for VPN on the performance measures like transfer rate, throughput, jitter, packet loss and etc.

The purpose of this project is to design SSL VPN between two computers using open source called Openvpn and evaluate its performance, to give this more secure solution good advance to replace most currently used solution IPsec VPN. We have been successfully developed and implement our design and the performance of this SSL VPN has been successfully tested.

It's found that in general the performance is not effected or effects is not noted with changing of encryption algorithm (each encryption algorithms has a different key size), except when measure include the header bits added for tunneling (tunneling: making VPN packet suitable with internet) like transfer rate and data access time of file which degraded with large key sizes e.g. AES key size (256 bit). Also there is a big difference in performance between encrypted data and unencrypted data.

And also as good edition to the project we make a simple test to measure RTT and throughput on IPsec VPN give high values in RTT as suggested (around 7.24 ms).

المستخلص

اكتسبت الشبكة المظهرية الخاصة أهتماماً متزايداً كتقنية لربط الشبكاتِ والمستعملين عن بعدٍ، هذا الإهْتِمام نشأ من منظورِ إقتصاديِ حيث ان الخطوط المؤجَّرةِ لَيسَ حَلّاً مناسباً إقتصادياً لربط شبكتين بعيدتينِ عن بعضهماَ ولا يَدْعمُ تداول محتويات الشبكات عن بُعْدَ من قبل المستخدمين وقد شجع علي انتشار الشبكة المظهرية الخاصة أنَّتشار استخدام الإنترنت في كل انحاء العالم. مع ذلك يَجِبُ أَنْ تكُونَ الشبكة آمنه بما يكفي بما اننا نستعملَ الإنترنتَ كوسيط ناقل للبيانات بين الشبكات حيث ان النترنت بيئة متاحة لجميع المستخدمين ، مما يجْعلُ أمر اتلاف البياناتِ أمراً سهلاً, إذاً لتكون الشبكة المظهرية الخاصة مكافئة لشبكة الخطوط المؤجرة يجب أن يتم دعمها بوسائل التشفير وكلمة المرور وموثوقية البيانات.معظم الشبكات المظهرية الخاصةِ تستخدمً بروتكول أمن بروتوكول الإنترنتِ (IPsec) لنقل البيانات بسرية عبر النترنت ، لكن هَلْ هو أفضل حَلّ ؟ نظام أمن الاتصالات (SSL) بروتكول هو البديل المرشح ليحل محلً بروتكول أمن بروتوكول الإنترنتِ برتكول، لكن ما هو مدى تأثير أستخدام بروتوكول نظام أمن الاتصالات علي كفاءة الشبكة المظهرية الخاصةُ.

ظهور الشبكة المظهرية الخاصة ذات نظام أمن الاتصالات (SSL VPN) يقودنا للتفكير في أجراء دراسات لمعْرِفة تأثير تشفيرِ البياناتِ و تاثير اضافة متطلبات حماية البياناتِ على أداءِ الشبكة مثل سرعة النقلِ، مدي النقل، فقدان حزم البيانات والخ.

يهدف هذا المشروعِ ألي تصمّمَ الشبكة المظهرية الخاصة ذات نظام أمن الاتصالاتَ بين حاسوبين باستعمال مصدرأًمفتوحاً برنامج يسمي Openvpn وتُقيّمُ أدائَها، مما يعطي هذا الحل) SSL VPN) دفعة ليحل محل أكثر الحَلّولِ المستعملِة حالياً (IPsec VPN). تم تصميم وتنفيذ شبكة SSL VPNبنجاح وقياس أداءَ هذه الشبكة.

وَجدنَا عُموماً أن التغير في اداء الشبكة لم يكن ملحوظاً مع تغير خوارزمية التشفير (كُلّ خوارزمة تشفير لَها حجم مفتاح مختلف) عندما يتضمن هذا المقياس تشفير البيانات فقط مثل مقياس الا نتاجية(Throughput) ، ولكنه تأثر عندما تضمّنُ المقياس البيانات المضافة لنقل وتأمين البيانات أي التي تَجْعلُ حزمةَ بيانات الشبكة الخاصة مناسبة للإنترنتِ مثل معدل النقلِ وزمنِ وصول بياناتِ الملفِ حيث تنخفض الكفاءة مع زيادة حجمِ مفتاح التشفير . ايضا يلاحظ أن قيمة نرفزة الحزم(بعدها من بعضها البعض jitter) هي قيم غير منتظمة, وهو ما يعرف ب النرفزة العشوائية .وأيضاً كأضافة مفيدة للمشروعِ تم اجراء إختبار بسيط لقيَأس وقت رحلة ذهاب وإيابِ (RTT) و (Throughput) الانتاجية على شبكة IPsec VPN .

Table OF Contents:

Table of Figures

List of Tables

Chapter 1

Introduction

1.1Overview

Due to the increasing number of corporations with many branches spread all over the world, it was necessary to find a way to minimize the cost needed to connect all these network branches .The conventional way to connect networks using dedicated lines (Leased lines, ranging from ISDN (integrated services digital network, 128 Kbps) to OC3 (Optical Carrier-3, 155 Mbps) fiber) is quite expensive and often rises in cost as the distance between the branches increases.

As the popularity of the Internet grew, businesses turned to it as a means of extending their own networks, which is the base of the technology known as virtual private network.

Basically, a VPN is a private network that uses a public network (usually the Internet) to connect remote sites or users together. Instead of using a dedicated, real-world connection such as leased line, a VPN uses "virtual" connection routed through the Internet from the company's private network to the remote site or employee and thus enabling them to work in a more flexible conditions.

The problem of using the internet is the security issue, when you are using the internet you are exposed and the traffic can be accessed easily. To solve this problem the concept of tunneling was introduced which can be done by many security protocols; one of them is SSL (secure socket layer). Using SSL to perform VPN tunneling is a new promising technology called SSL VPN.

1.2 Problem Definition:

Virtual private network tunneling is done using many security protocols such as IPsec, L2TP, PPTP, and SSL.

IPsec tunneling protocol is the most used protocol for VPN tunneling because when they designed VPNs it was the only protocol for this purpose, but one of major gripes

with IPsec is that it adds a lot of complexity to the kernel. Complexity is really the enemy of security. Also when using IPsec we need client software on each PC from which a user needs access; an IPec VPN may also require changes to the desktop. These factors result in high support costs.

SSL VPNs use a different methodology to transport private data across the public Internet. Instead of relying upon the end user to have a configured client on a company laptop, SSL VPNs use SSL/HTTPS, which is available without additional software deployment on all standard Web browsers, as a secure transport mechanism. Using an SSL VPN, the connection between the mobile user and the internal resource happens via a Web connection.

SSL is located as a layer after the application layer, as opposed to IPsec VPNs' open "tunnel" at the network layer. The use of SSL is ideal for the mobile user because:

SSL VPNs do not require a client download onto the device being used to access corporate resources.

SSL operates at the application layer, independent of any operating system, so changes to the OS do not require an update in the SSL implementation. And because SSL VPNs operate at the application layer, it is possible to offer extremely granular access controls to applications, making it ideal for mobile workers and those users coming from an unmanaged or entrusted end-point. So SSL have more advantages than IPsec.

It provides many techniques to provide security, but are the techniques having effects in the performance of the network?

1.3 Objectives:

- Real time implementation of a SSL VPN between two computers.
- Performance evaluation of the implemented SSL VPN.

1.4 TOOLS:

The design of the system will be software implementation of SSL VPN network support features make the performance evaluation be easy to be achieved like File Transfer Protocol FTP and Network File System Protocol NFS and performance measuring protocol. It was developed and implemented over Linux RED HAT 4 operating system into two computers have net access using the following tools:

- Openvpn software: Open source software used to implement full SSL VPN.
- Netperf, Netserver ,Iperf and TTCP rpms software for performance measuring of SSL VPN.
- NFS, FTP rpms software to support protocol.
- Jperf software for IPsec performance measuring.
- Statistical Package for the Social Sciences (SPSS) software: for statistic analysis of performance results.

1.5 Thesis Layouts

This thesis is organized as follows:

Chapter 2: Introduces the concepts about virtual Private Network's theory and tunneling protocols with more details about SSL VPN.

Chapter 3: The design and system implementation are comprehensively explained.

Chapter 4: Describes, how performance evaluation implemented and the result gained with analysis.

Chapter 5: Presents conclusions and recommended future work.

Appendix A: Openvpn configuration steps with clear description.

Appendix B: Openvpn configuration's file with description1.5 Thesis Layouts

Chapter 2

Literature Review

2.1 A clear Division between Public and Private Networks:

A public network, like the public telephone system and the Internet, is a large collection of unrelated peers that exchange information more or less freely with each other. The people with access to the public network may or may not have anything in common, and any given person on that network may only communicate with a small fraction of his potential users.

A private network is composed of computers owned by a single organization that share information specifically with each other. They're assured that they are going to be the only ones using the network, and that information sent between them will (at worst) only be seen by others in the group. The typical corporate Local Area Network (LAN) or Wide Area Network (WAN) is an example of a private network.

A virtual private network is a way to simulate a private network over a public network, such as the Internet. It is called "virtual" because it depends on the use of virtual connections that is, temporary connections that have no real physical presence, but consist of packets routed over various machines on the Internet.

2.2 Private Networks through Internet

The world has changed a lot in the last couple of decades. Instead of simply dealing with local or regional concerns, many companies have facilities spread out across the country or around the world, and there is one thing that all of them need a way to maintain fast, secure and reliable communications wherever their offices are.

Until fairly recently, this has meant the use of leased lines to maintain a wide area network (WAN). Leased lines, ranging from ISDN (integrated services digital network, 128 Kbps) to OC3 (Optical Carrier-3, 155 Mbps) fiber, provided a company with a way to expand its private network beyond its immediate geographic area. A WAN had obvious advantages over a public network like the Internet when it came to reliability, [1]

performance and security. But maintaining a WAN, particularly when using leased lines, can become quite expensive and often rises in cost as the distance between the offices increases. The most popular use of VPN is in national companies which have main offices outside Sudan (MTN, ZAIN, Petronas).

As the popularity of the Internet grew, businesses turned to it as a means of extending their own networks. First came intranets, which are password-protected sites designed for use only by company employees. Now, many companies are creating their own VPN **(**virtual private network**)** to accommodate the needs of remote employees and distant offices.

Basically, a VPN is a private network that uses a public network (usually the Internet) to connect remote sites or users together. Instead of using a dedicated, real-world connection such as leased line, a VPN uses "virtual" connections routed through the Internet from the company's private network to the remote site or employee. This can be done by using various *tunneling* protocols and by encrypting, decrypting and authenticating traffic. [1]

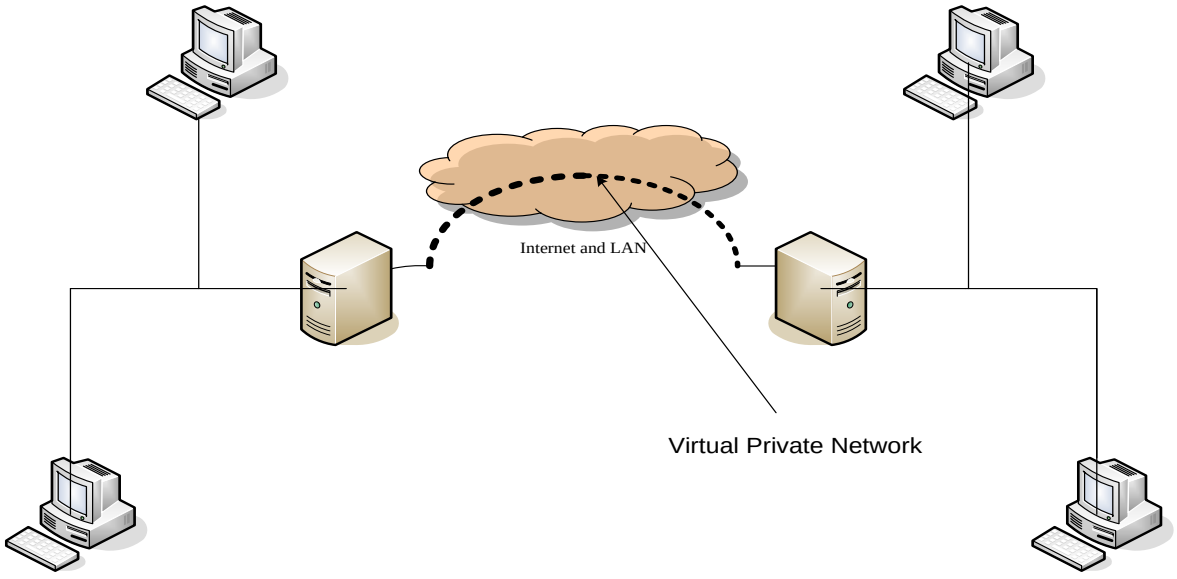

Figure 2.1 Virtual Private Network VPN

2.3 Types of VPN:

There are two main types of VPN remote access VPN and site-to-site VPN

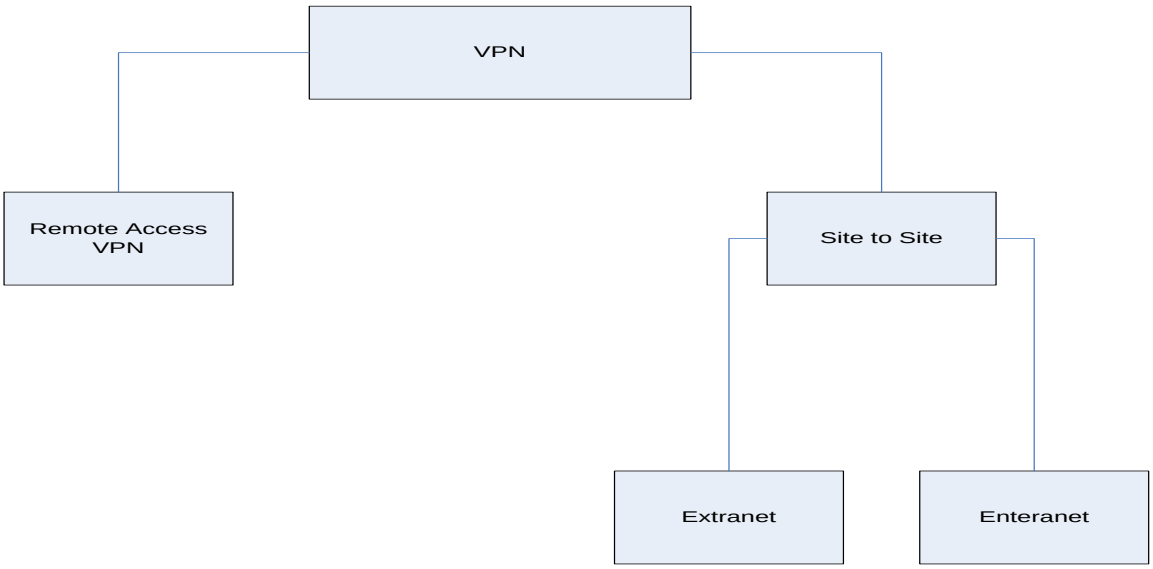

Figure 2.2 VPN Types

1. Remote-Access

Also called a virtual private dial-up network (VPDN) is a user-to-LAN connection used by a company that has employees who need to connect to the private network from various remote locations.

A good example of a company that needs a remote-access VPN would be a large firm with hundreds of sales people in the field. Remote-access VPN permit secure, encrypted connections between a company's private network and remote users through a third-party service provider.

2. Site-to-Site VPN

Through the use of dedicated equipment and large-scale encryption, a company can Connect multiple fixed sites over a public network such as the Internet. Site-to-site VPNs can be one of two types:

- **Intranet-based** - If a company has one or more remote locations that they wish to join in a single private network, they can create an intranet VPN to connect to LAN.

- **Extranet-based** - When a company has a close relationship with another company (for example, a partner, supplier or customer), they can build an extranet VPN that connects LAN to LAN, and that allows all of the various companies to work in a shared environment. [2]

2.4 Tunneling Concept

Tunneling is a way of forming a virtual network on top of a physical network. The data to be transferred (payload) can be the frames (or packets) of anther protocol, instead of sending a frame as it is produced by originating node, the *tunneling protocol* encapsulates the frame in an additional header. The additional header provides routing information so that the encapsulated payload can traverse intermediate network

- Tunneling process include :
  - Encapsulation
  - Transmission
  - Unencapsulation
  - And security must be available since the Internet is a public network, you always risk having someone access any system you connect to it.

2.5 Tunneling protocols:

For tunneling to be established, both the tunnel client and tunnel server must be using the same tunneling protocol. Tunneling Protocols can be Layer 2(OSI model) or Layer3 protocols.

a) Layer 2 Protocols: correspond to the data link layer and use frames as their unit of exchange, Point to Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP) are layer 2 protocols.

b) Layer 3 protocols: correspond to the network layer, and use packets as their unit of exchange, IP security (IPSec) at *tunneling mode* is layer 3 tunneling protocol, it encapsulate IP packets. [3]

2.6 Encryption

Encryption is the process of taking all the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Most computer encryption systems belong in one of two categories:

❖ Symmetric-key encryption: both computers have a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to another computer.

❖ Public-key encryption: Public-key encryption uses a combination of a private key and a public key. The private key is known only to your computer, while the public key is given by your computer to any computer that wants to communicate securely with it. To decode an encrypted message, a computer must use the public key, provided by the originating computer, and its own private key.

2.7 Digital signature

Other services we expect from a secure system are authentication and message integrity message authentication means that the receiver needs to be sure of the sender's identity. Message integrity means that the data must arrive at the receiver exactly as they were sent. This to services can be provided by Digital Signature: the sender signs a digest of the message generated by Hash functions.

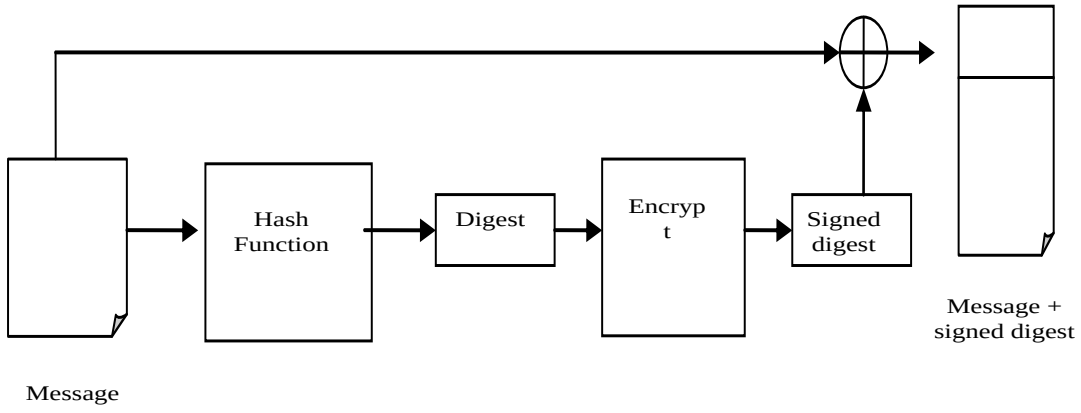

Figure 2.3 Digital Signature

2.8 IP Security Protocol (IPsec)

Internet Protocol Security Protocol (IPsec) provides enhanced security features such as better encryption algorithms and more comprehensive authentication. IPsec has two encryption modes:

a. **Tunnel**: encrypts the header and the payload of each packet.

b. **Transport**: only encrypts the payload .The original routing information in the packet is not protected.

Only systems that are IPsec compliant can take advantage of this protocol. Also, all devices must use a common key and the firewalls of each network must have very similar security policies set up. IPsec can encrypt data between various devices, such as: Router to router, Firewall to router, PC to router, PC to server[4]

2.9 Secure Socket Layer (SSL)

For many years, IPsec was the only security protocol available to secure the site-to-site or client-to-server VPNs. By chance, this is now changed with the release of the SSL protocol. Available at the beginning to secure specific protocols like HTTP, SSL is now able to secure any application and encrypt TCP or UDP tunnels to create site-to-site or client-to-site VPNs.

2.9.1 SSL DESCRIPTION:

SSL (for Secure Sockets Layers) has been created by Netscape in the 90s. Two SSL versions have been released v2 (1994) and v3 (1995). The patent was then bought and updated by the IETF in 2001. At the same time it was renamed as TLS which stands for transport layer security. The SSL word is commonly used to designate both the SSL and TLS protocols. The last version of TLS is v1.1. [5]

2.9.2 SSL GOALS:

The two main SSL goals are the following:

a. Authenticate the server and the client using the Public Key Infrastructure (PKI).
b. Provide an encrypted connection for the client and server to exchange messages.

2.9.3 OpenVPN AND THE OSI MODEL:

Where could we place the SSL protocol in the OSI model? The standard OSI model is composed of seven layers while a four layer model matches more closely the TCP/IP architecture used by the large majority of the applications.

SSL is located between the application and transport layers and will encrypt the application layer.

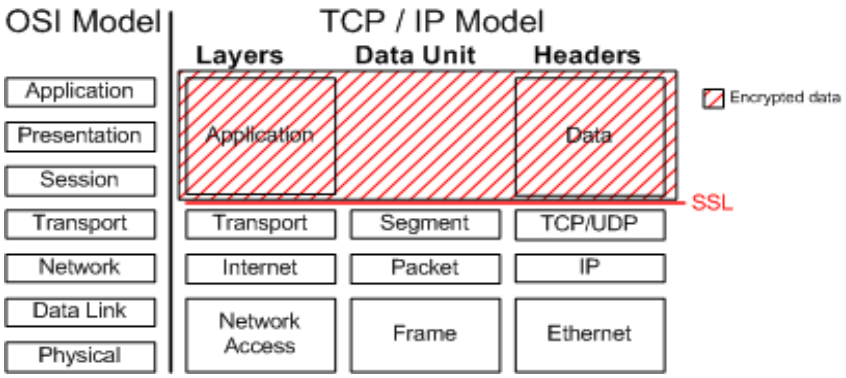

Figure 2.4 SSL Layer in OSI Model and TCP/IP Model

2.9.4 TYPE OF SSL VPN:

In the past, SSL was a protocol used with specific applications like HTTP; however for some years able it has been able to potentially secure the transactions of any applications over Internet and to create encrypted tunnel (VPNs) in the same manner than IPsec does. Two types of VPNs are available:

- Client-to-server (or remote access) VPNs where the client needs a web browser such as Firefox.
- Site-to-site where a specific software is required such as OpenVPN[4]

2.9.5 SSL VPN vs. IPsec VPN

The goal of SSL and IPsec is the same: create VPNs and thus encrypt traffic between two devices with the same algorithms. But as you will see, the way to accomplish this task is very different. These differences displayed into below table.

Table 2.1 Comparison between SSL and IPsec

	IPSEC	SSL
RFC:	2401	4346 (TLS 1.1)
OSI position:	Internet Layer	Between Transport and Application Layers
Software location:	Kernel space	User space
Installation:	Vendor non-specific	Vendor specific
Configuration:	Complex	Simple
NAT:	Problematic	No problem
Firewall:	Not friendly	Friendly

Chapter 3

Design and Implementation

This chapter presents the design of the VPN, its procedures and elements (Platform, software, hardware components) of the implementation.

First, goals of the implementation idea will clarified to get an overview about the design.

Second, start overview to general design some details about main elements of this general idea like platform and software's.

Third go more deep into the design procedures and elements of VPN, more about software procedure and more about protocols configurations that assist in achieve goals of implementation

3.1 Goals of the Design

VPN implementation is the first step to measure the advantages of SSL VPN using performance evaluation manners of networks in general which will be shown later.

So the implementation will be sufficient to perform performance evaluation tests .Also implementation is a goal in it self, it give a good experience in performing virtual private network.

Implementation must support VPN requirements:

a. User authentication : verify the user identity

b. Address management : assign client's address and address privacy

c. Data encryption

d. Key management: generate and refresh encryption keys for the client and the server.

e. Multi-protocol support

f. Software dependant

After the implementation of the VPN, the performance will be measured; performance measurement means the effects of encryption in the performance measures of network (Throughput, Jitter, Round Trip Time (RTT), and Packet Loss) and the effects will be checked through different encryption algorithms.

3.2 VPN Design Overview

The basic idea of the implementation is achieving VPN network between two computers have net access using SSL tunneling protocol. It will be software implementation; one computer is configured as server and another one as client, each of the two computers is a part of LAN which will route client's packets (server's packets) to internet which definitely route it to its destination here server (client) through the server LAN (client LAN) As shown in figure 3.1, the VPN solution (design) must satisfy the Goals specified in previous section, which support sufficient security level.

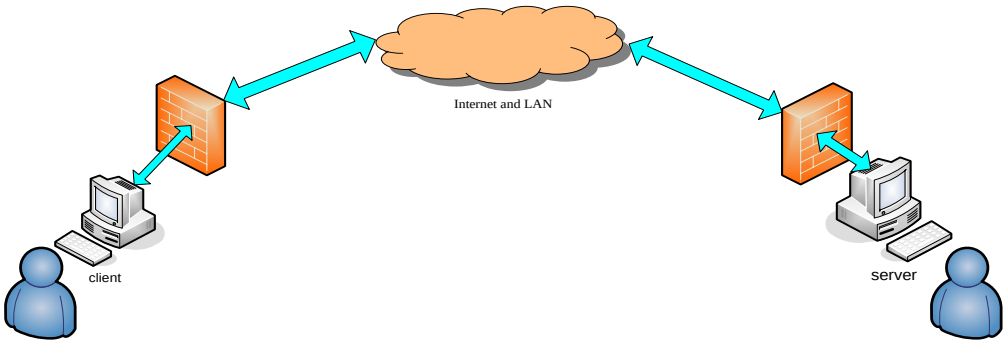

Figure 3.1 VPN Design Overview

To configure SSL VPV, open source software called Openvpn is installed in both the client and the server devices, both devices are based on LINUX operating system which is more suitable for security.

3.2.1 Platform: Linux RED HAT 4

Why Linux?

1. More secure against hackers (most of viruses are in windows platform)
2. Openvpn installation easier in Linux than windows platform

3.2.2 Software: OPENVPN

True SSL VPNs are beginning to appear in the market. One of the best, and definitely the least expensive, is the open source SSL VPN called Openvpn. Openvpn is highly user space (it does not require sophisticated intertwining with the OS's kernel to function) flexible tunneling application. It supports SSL/TLS security, Ethernet bridging, TCP or UDP tunnel transport through proxies or NAT, support for dynamic IP addresses and DHCP, scalability to hundreds or thousands of users, and portability to most major OS platforms. Openvpn currently runs on most OS's including windows 2000/XP, Linux, Solaris, BSD, and Mac OS X. Since it runs in user-space instead of as a kernel module, installation is so easy. [6]

3.2.2.1Openvpn Configuration:

Openvpn is configured like most UNIX services using a config(configuration) file. One of the blessings of Openvpn is the fact that the config. File format is almost exactly the same for all platforms.

We will display the steps performed to configure Openvpn in server and in client devices in below Block diagram, to more information see appendix A:

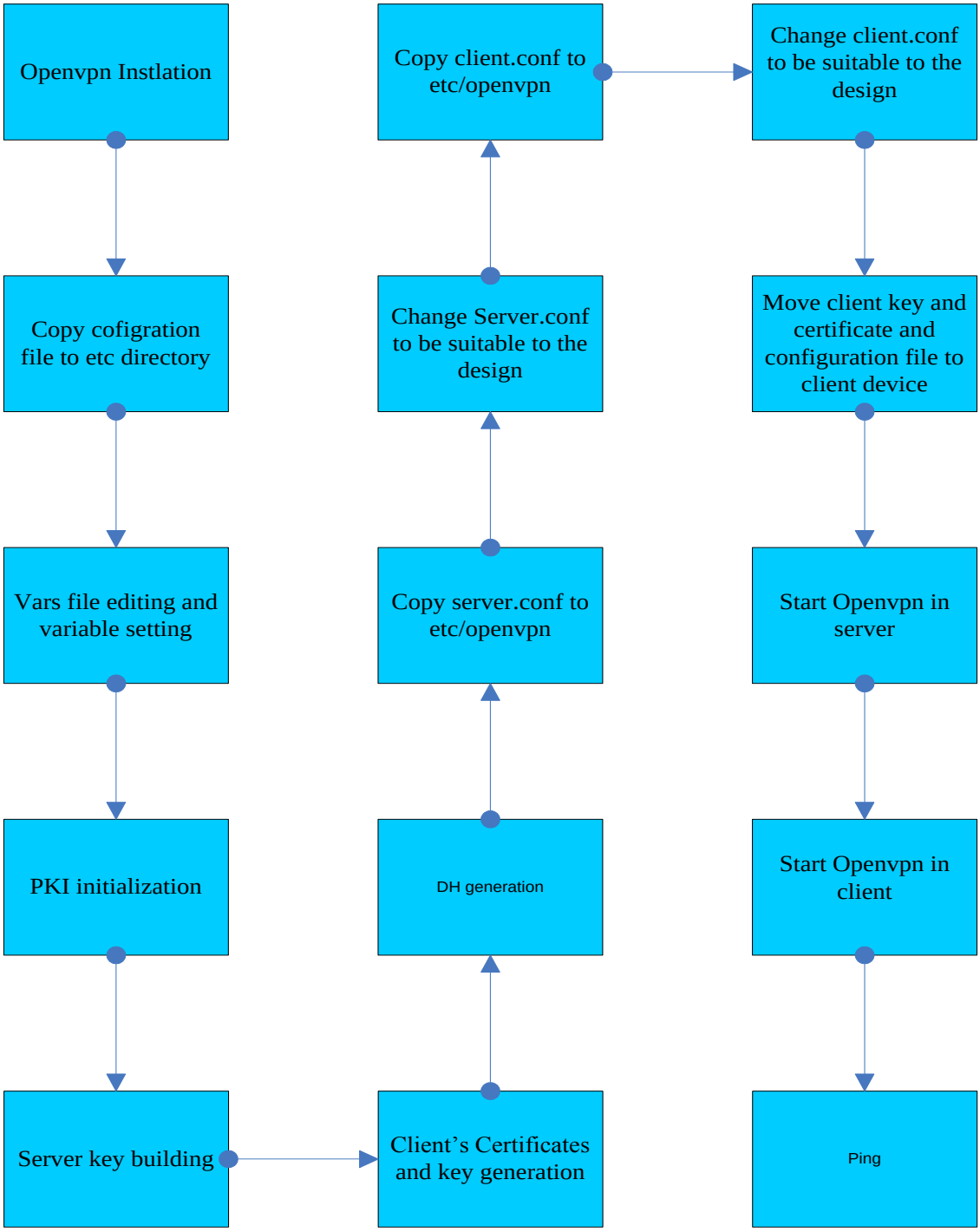

Figure 3.2 Openvpn Configuration Steps

3.2.2.2 Openvpn configurations file in details

In this section some details about settings of OPENVPN configuration file which given in appendix B, and clarify why we prefer this setting.

a. Tunnel mode

Although bridges and routers perform the similar task of delivering packets of data to their intended destinations, the network implications of choosing a routed or bridged architecture are significant. [4]

Bridging**:** In small networks bridges provides some benefits like

- Protocol transparency: the ability to support many protocols.
- Ease of setup: because a bridge uses little intelligence in making its network decisions, it is very easy to install and maintain.

Bridging networks learn where packets should go by using a broadcast, so if we send to new user the bridge will broadcast the packets to all users and only the new user will accept these packets. Bridge works on layer 2 on OSI model so it doesn't used to make connection between two different networks (i.e. IPX with IP). Bridges can find devices in a connected network because device addresses are carried in layer two.

Routing**:** deal with packets of information. Most importantly, routers operate in the Network Layer of the OSI model, routers can provide various schemes of filtering, path control and traffic control functions, also can handle one or more protocols such as TCP/IP, IPX, etc. so we can use it to connect two different networks (i.e. IPX with IP)

Generally routing is better than bridging.

When we use bridging?

Use bridging when you cannot subnet your IP network, and when you need to use non-routable protocols such as NetBIOS, or DECnet.

b. Tunnel port

Default source and destination tunneling port is UDP 1194, you may need to change it for Firewall reasons otherwise you can keep it. Prefer UDP ports. The use of TCP can lead to degraded performances.

As the majority of the applications use TCP, if you choose TCP tunneling, you will create a TCP over TCP tunnel. This is not recommended because in case of packets retransmissions on the interior TCP tunnel, recomputation will occur in both tunnels leading to slow performances such as high response time. Thus, prefer the UDP protocol to tunnel your application since contrary to TCP. [4]

If Firewall holds 1194 UDP port idle, solution one of follows:

- Change it to another UDP port
- Firewall setting must be changed to allow 1194 UDP port to work right.
- Firewall disable using this command (in linux):

```
# Service iptables stop
```

c. Hash function algorithms

The Integrity uses hash function algorithms to protect the data from being altered. HMAC **Keyed-Hash Message Authentication Code**, or **HMAC**, is a type of message authentication code (MAC) calculated using a specific algorithm involving a cryptographic hash function in combination with *a secret key* Any iterative cryptographic hash function, such as MD5(Massage Digest Algorithm 5) or SHA-1(Secure Hash algorithm), may be used in the calculation of an HMAC. The OpenVPN default hash functions are HMAC-SHA1. [7]

The hash algorithm has important property which feed into the algorithm is the hash size. For example MD5 has a hash size of 128 bits (16 bytes) and SHA-1 has hash size of 160 bits .How we generate a secret key? Using Deffie-*hellmann (DH)*

d. Diffie-Hellmann (DH) Settings

DH is protocol give one-time session key for two parties (e.g. two computers) the two Computers use the session key to exchange data without having to remember or store it for future use. So the symmetric key is used only once. Once the OpenVPN peers are sure about each other's identity, DH can be used to create a shared secret key for the hash function and the cipher algorithm.[2]

e. Cipher Algorithms

The confidentiality is ensured with symmetric ciphers such as 3DES or AES to protect the data from being read. The OpenVPN default cipher algorithm is Blowfish.

`BF-CBC`	`Blow fish`	Cipher Block Chaining (128 bit)
`AES-256-CBC`	`Advanced 256 key`	Cipher Block Chaining(256 bit)
`DES-EDE3-CBC`	`Triple-DES`	Cipher Block Chaining (122 bit effective key)

CBC: **C**ipher **B**lock **C**haining is a cryptography operational mode used to encrypt data with a cipher block algorithm like the AES, DES or Blowfish.

g. Public Key Infrastructure (**PKI**) is an arrangement that binds public keys with respective user identities by means of a certificate authority (CA). The user identity must be unique for each CA. The binding is established through the registration and issuance process. The PKI role that assures this binding is called the Registration Authority (RA). For each user, the user identity, the public key, their binding, validity conditions.[8]

h. IP Addressing

VPN IP addresses must be at difference address domain from the local network e.g. university LAN has 172.15.0.0 domain there for we use 10.8.0.0 domain to avoid conflict.

OpenVPN can handle an IP address pool dynamically by itself using the "server" directive on a new subnet, which is not currently being used. Or use ifconfig-push to assign static IP addresses by certificate common name, but again, use fresh IP addresses outside of any LAN subnets in use.

As shown in configuration file (in appendix A) we use ‘server’ directive to generate IP address 10.8.0.1 for server and the client (we just have one client) IP address generated dynamically in same domain (default 10.8.0.6).

3.2.3 FTP and NFS configuration

As complement factor of VPN network implementation we configure file transfer protocol (FTP) and Network File System (NFS) protocols which help us in performance evaluation, exactly beside using software techniques to measure transfer rate we use FTP to transfer file and get the real rate of this file transferring, also NFS help us to measure access time of mounting file in the server device from the client device.

a. FTP: the File Transfer Protocol (FTP) is a network protocol used to transfer data from one computer to another through a network, such as over the VPN or internet.

As shown in Figure 3.2 we use FTP to get test-file from server. After the execution the FTP displays a summery of the process contain the transfer rate (all result at Chapter 4).

b. NFS: Network File System (NFS) is a network file system protocol, allowing a user on a client computer to access files over a network as easily as if the network devices were attached to its local disks.

Figure 3.3 file transferring using FTP and file mounting using NFS

As shown in Figure 3.3 NFS use to mount Test-file from the server and Tcpdump which track packets between server and client will display a time difference between a sending request And acceptation acknowledge(all result at Chapter 4). So to calculate the time difference the two devices time must synchronize, we use NTP protocol to adjust devices at same network time through network. [9]

c. NTP: The Network Time Protocol (NTP) is a protocol for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks. NTP uses UDP port 123 as its transport layer.

We use NTP interface to synchronize our client has 10.8.0.6 IP address to

The server has 10.8.0.1 IP address.

3.2.4 Hardware Used in the Design

in this section the hardware used will be described in some details , the implementation of system require two computers and , and public network (internet access) , so we will clarify specifications of computers have been used and internet service which has real effect in the performance of the VPN as will be discuss in next chapter .

3.2.4.1 Computers Specifications

The specification of computers will contain the RAM capacity, the speed of CPU and the kernel, to both server and client

- RAM: 512 MB,
- CPU information : shown in below table

Table 3.1 CPU Information

Characteristic	Server	client
Speed	2.66 GHz	2.8 GHz
vendor_id	GenuineIntel	GenuineIntel
Model name	Intel(R) Pentium(R) 4 CPU 2.66GHz	Intel(R) Pentium(R) 4 CPU 2.66GHz
Cache size	512 MB	240 MB

- Kernel: Kernel 2.6.9-5.EL

3.2.4.2 Internet Service Specifications

The VPN implementation is performed in Electrical engineering department Labs, and the performance of VPN is differ between two states, this states depend on two factors, will be describing bellow:

- Factor one :Change of The NET Service

The specification of UofK center campus internet changed during the performance evaluation of the VPN which in general make the network have better speed than Pervious.

Old Specifications:

The center campus is feed from 1.5 Mbit/second HDSL which shared between all faculties include engineering campuses.

New Specifications:

The university network is feed by 5Mbit/second using fiber optic cable directly from Data Net and then shared between all campuses.

- Factor two: Change Tests Location Faculty

This change may have an effect on VPN performance because first lab uses Intel Express Hub and the anther lab use switch which is higher speed.

Chapter 4

Performance Evaluation

In this chapter we will clarify some parameters (output parameters) measured to describe our SSL VPN network, and what software (tools) used to measure them and what the input parameters we change it in network characteristics to describe the change in network behavior according to it.

4.1 Testing Parameters and Measures

The network algorithms that describe the cryptography settings called here Input Parameters and the measures that describe the performance of VPN called Output Measures:

4.1.1 Input Parameter

They are those cryptography algorithms that used in SSLVPN network to tunnel security Issues like *Ciphers algorithms* and *Hash function algorithms* whish described in previous chapter, so easily we change this input parameter in Open VPN configuration file and measure Output Parameter using some software, the main purpose is to know the effect of cryptography algorithms on network performance.

4.1.2 Output Measures

They are those network parameters that used to describe the performance of each network like jitter, throughput, access time, round trip time, packet loss, and bandwidth. Here we will define each term of them.

- Jitter: the variation in the time between packets arriving caused by network congestion, timing drift, or route changes.[10]
- Throughput: is a measure of the amount of data that actually sent (with out header) over a link in a given amount of time. It effected by broadcast traffic ,collisions, routing protocols.[11]
- Access Time: the time that taken by client to access a file in server.

- Round Trip Time (RTT): Round Trip Time, it is a measure of the time it takes for a packet to travel from a computer, across a network to another computer, and back.
- Packet Loss: number of packets lost on a connection (i.e. Ping connection)
- Bandwidth: The term *bandwidth* is typically used to describe the capacity of a link. For our purposes, this is the maximum transmission rate for the link. If we can transmit onto a link at maximum10 Mbps, then we say we have a bandwidth of 10 Mbps. Its unit is (bit/sec).[11]
- Transfer rate : the number of bits that can be transferred across a network connection in one second; commonly used as a synonym for throughput or bandwidth; technically, it is always greater than throughput because extra bits are included in each data packet for routing purposes.

4.2 Testing Mechanisms:

Testing mechanisms is the commands or software used to measure performance parameters, we use:

- Iperf : comes from the National Laboratory for Applied Network Research (NLANR)

  Measure: Jitter and Bandwidth

  Example: of Bandwidth. Cipher: AES, Hash: MD5

Server side

```
[root@localhost ~]# iperf -s –p1194
------------------------------------------------------------
Server listening on TCP port 1194
TCP window size: 16.0 Kbytes (default)
------------------------------------------------------------
[ 4] local 10.8.0.6 port 3000 connected with 10.8.0.1 port 1194
[ ID] Interval     Transfer       Bandwidth
```

```
[ 4] 0.0-10.0 sec  182 MBytes   152 Mbits/sec
```

Client side

```
[root@localhost ~]# iperf -c10.8.0.1 -p1194
------------------------------------------------------------
Client connecting to 10.8.0.1, TCP port 1194
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 10.8.0.6 port 32784 connected with 10.8.0.1 port 1194
[  3]  0.0-10.0 sec    182 MBytes    152 Mbits/sec
```

- **Netperf**: Another program to consider is *netperf*, which had its origin in the Information Networks Division of Hewlett-Packard. It is freely available, runs on a number of Unix platforms, it supports a much wider range of tests.[12]

  Unlike with *ttcp*, the client and server are two separate programs. The server is *netserver* the client is known as *netperf*.

  Measure: Throughput (bit/s).

  Example: of measuring throughput. Cipher: AES, Hash: MD5

Server side

```
[root@localhost ~]# netserver
 Starting netserver at port 1194
```

Client side

```
[root@localhost ~]# netperf -H10.8.0.1 -p1194
 TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to
10.8.0.1 (10.8.0.1) port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
```

```
Size    Size     Size      Time      Throughput
bytes   bytes    bytes     secs.     10^6bits/sec

87380   16384   16384     10.00       47.89
```

- **TTCP**: One of the oldest bulk capacity measurement tools is *TTCP*. This was written by Mike Muuss and Terry Slattery. To run the program, you first need to start the server on the remote machine using, typically, the *-r* and *-s* options. Then the client is started with the options *-t* and *-s* and the hostname or address of the server. Data is sent from the client to the server, performance is measured, the results are reported at each end, and then both client and server terminate. [12]

  Measure: Throughput (B/s).

  Example: of measuring throughput Cipher: AES Hash: MD5

  Server side

```
[root@localhost ~]# ttcp -r -s
ttcp-r: buflen=8192, nbuf=2048, align=16384/0, port=1194
ttcp-r: sockbufsndsize=16384, sockbufrcvsize=87380,
sockbufsize=51882,  # tcp #
ttcp-r: accept from 10.8.0.6
ttcp-r: 16777216 bytes in 3.665 real seconds = 4470.974 KB/sec
+++
ttcp-r: 12464 I/O calls, msec/call = 0.301, calls/sec = 3401.258
ttcp-r: 0.011user 0.100sys 0:03real 3% 0i+0d 0maxrss 0+3pf
12463+2csw
```

  Client side

```
[root@localhost ~]# ttcp -t -s 10.8.0.1
ttcp-t: buflen=8192, nbuf=2048, align=16384/0, port=1194
ttcp-t: sockbufsndsize=16384, sockbufrcvsize=87380,
sockbufsize=51882,           # tcp -> 10.8.0.1 #
ttcp-t: connect
```

```
ttcp-t: 16777216 bytes in 3.661 real seconds = 4475.264 KB/sec
+++

ttcp-t: 2048 I/O calls, msec/call = 1.831, calls/sec = 559.408

ttcp-t: 0.000user 0.083sys 0:03real 2% 0i+0d 0maxrss 0+4pf
463+4csw
```

- Ping: is a computer network tool used to test whether a particular host is reachable across an IP network; it is also used to self test the network interface card of the computer. It works by sending ICMP "echo request" packets to the target host and listening for ICMP "echo response" replies. Ping estimates the round-trip time, generally in milliseconds, and records any packet loss, and prints a statistical summary when finished

  Measure: RTT and Packet loss

  Example: of RTT. Cipher: AES, Hash: Md5

  At client:

  [root@localhost ~]# ping 10.8.0.1

  PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.

  64 bytes from 10.8.0.1: icmp_seq=0 ttl=64 time=0.797 ms

  64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.692 ms

  64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.693 ms

  64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=0.662 ms

  64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=0.671 ms

  64 bytes from 10.8.0.1: icmp_seq=5 ttl=64 time=0.675 ms

  64 bytes from 10.8.0.1: icmp_seq=6 ttl=64 time=0.680 ms

  64 bytes from 10.8.0.1: icmp_seq=7 ttl=64 time=0.679 ms

  64 bytes from 10.8.0.1: icmp_seq=8 ttl=64 time=0.682 ms

  64 bytes from 10.8.0.1: icmp_seq=9 ttl=64 time=0.673 ms

```
--- 10.8.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 6.904ms
rtt min/avg/max/mdev = 0.662/0.695/.797/0.072 ms, pipe 2
```

- **Tcpdump:** is a common computer network debugging tool that runs under the command line. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over network to which the computer was attached.

4.3 Test Result and Analysis using SPSS of SSL VPN

This section, to show the results and the analysis's done by SPSS (Statistical Package for the Social Sciences is a computer program used for statistical analysis) which content calculations of mean, standard deviation, maximum and minimum value and variance, and comparison graphs and conclusion about the effect of changing cryptographic algorithms on the performance.

4.3.1 Iperf Bandwidth Results

Below results of VPN bandwidth (Mbit\sec) for different cipher algorithms and hash functions algorithms

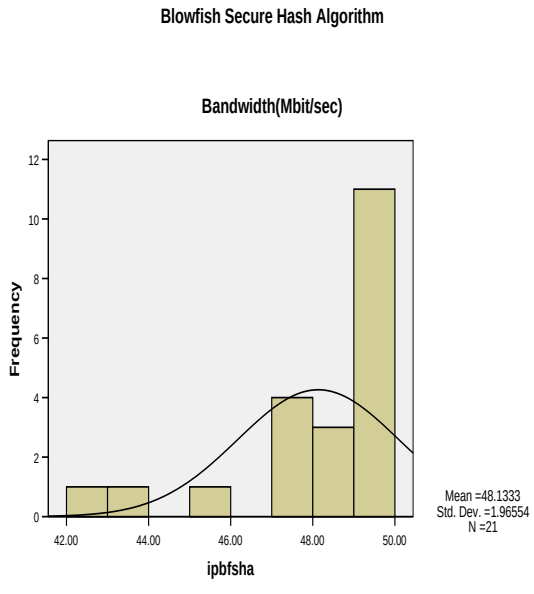

Figure 4.1 BF-SHA Bandwidth

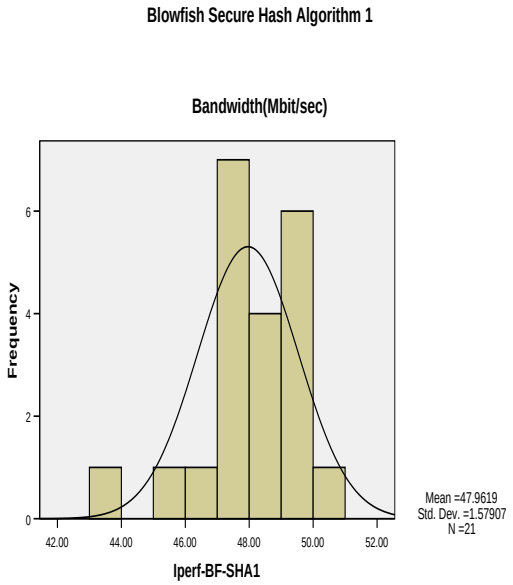

Figure 4.2 BF-SHA1 Bandwidth

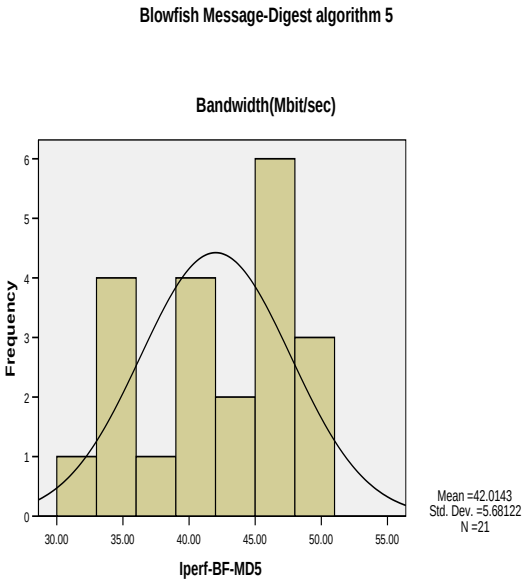

Figure 4.3 BF-MD5 Bandwidth

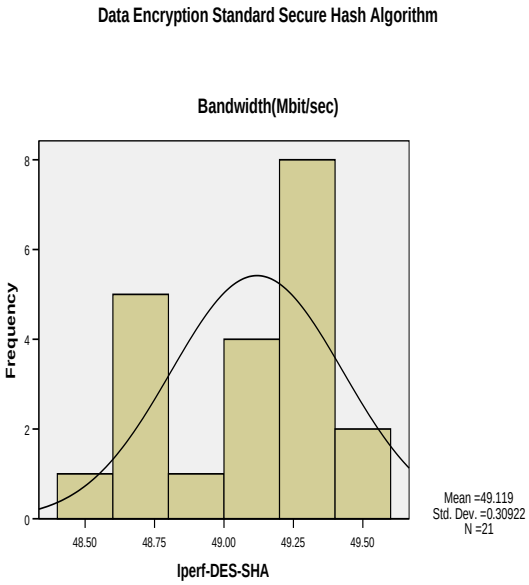

Figure 4.4 DES-SHA Bandwidth

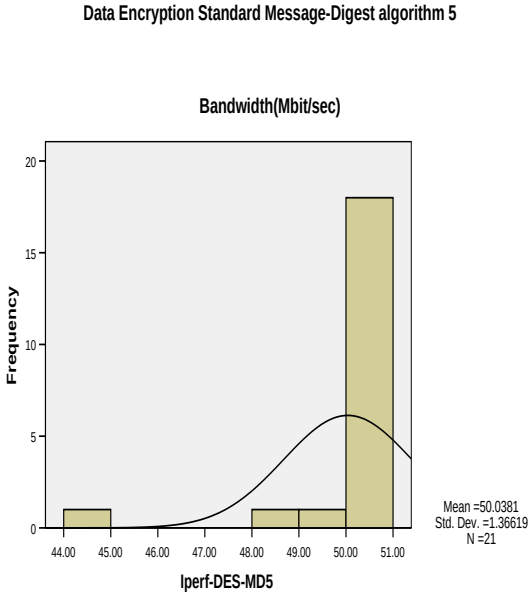

Figure 4.5 DES-MD5 Bandwidth

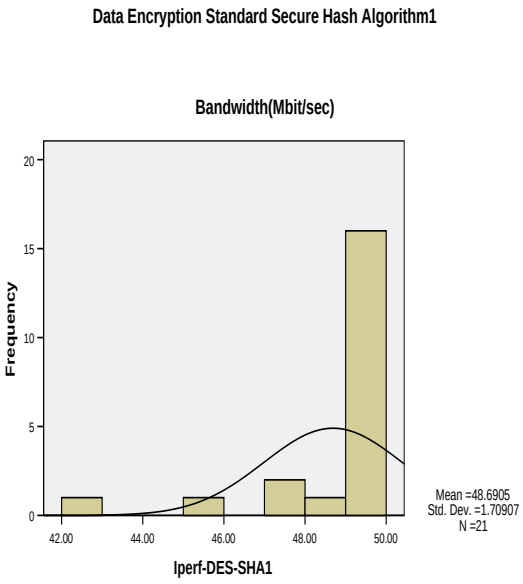

Figure 4.6 DES-SHA1 Bandwidth

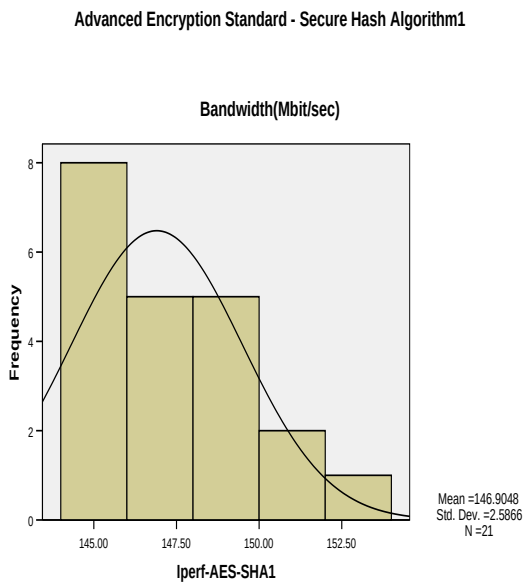

Figure 4.7 AES-SHA1 Bandwidth

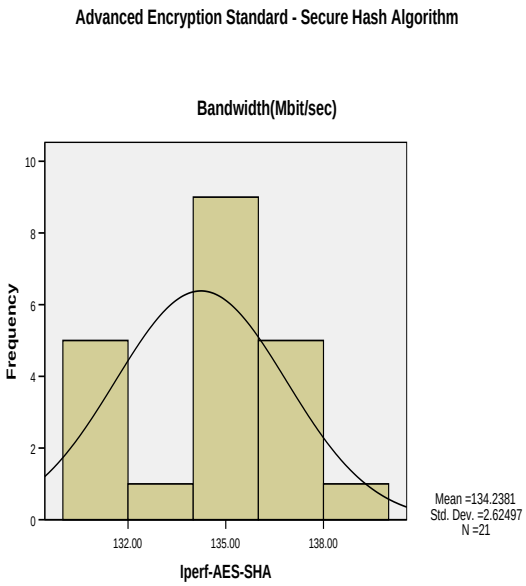

Figure 4.8 AES-SHA Bandwidth

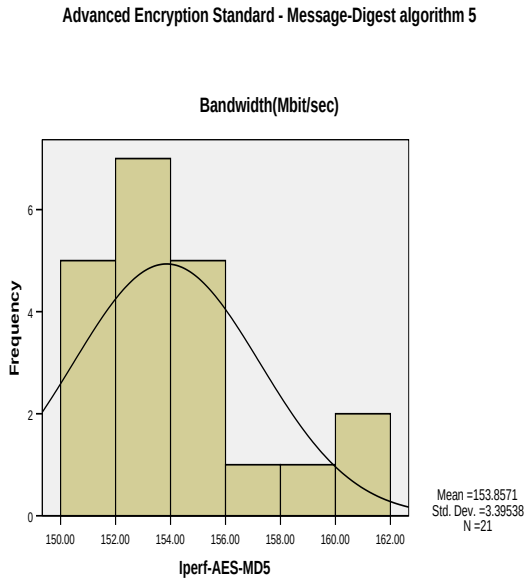

Figure 4.9 AES-MD5 Bandwidth

TABLE 4.1 Bandwidth Statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
Bf-sha	48.13	49.70	42.80	1.97	3.86
Bf-sha1	47.96	50.30	43.20	1.58	2.49
Bf-md5	42.01	50.20	32.10	5.68	32.28
Des-sha	49.12	49.50	48.50	.31	.10
Des-sha1	48.69	49.50	42.40	1.71	2.92
Des-md5	50.04	50.70	44.50	1.37	1.87
Aes-sha	134.24	140.00	130.00	2.62	6.89
Aes-sha1	146.90	153.00	144.00	2.59	6.69
Aes-md5	153.86	162.00	150.00	3.40	11.53

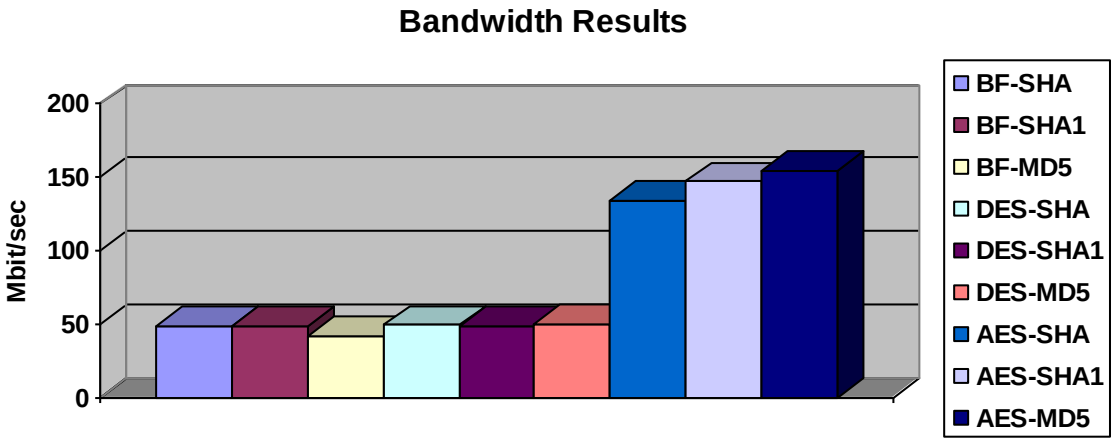

Fig 4.10 Bandwidth Mean Values

from above figures and table the bandwidth is not significantly change through all the algorithms except at AES values, which large difference acquire but the real reason as mentioned in section 3.2.4, that this values are taken after change in net service from 1.5 Mbit/s to 5 Mbit/s with fiber optic cable, and the location of the implementation is changed to another lab use switch instead of hub used in previous lab. So the Bandwidth doesn't change with encryption and Hash algorithms.

4.3.2 Netperf Throughput Results

Below results of throughput (Mbit\sec) for different cipher algorithms and hash functions algorithms.

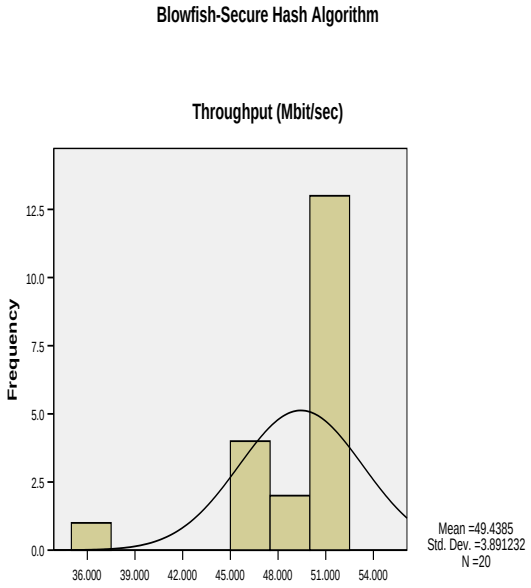

Figure 4.11 BF-SHA Throughput

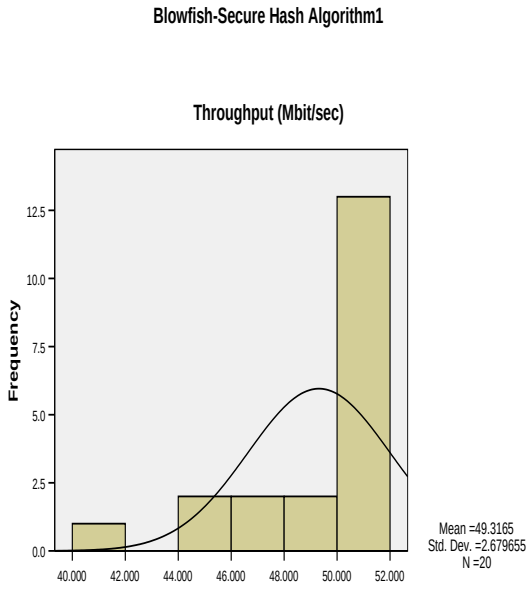

Figure 4.12 BF-SHA1 Throughput

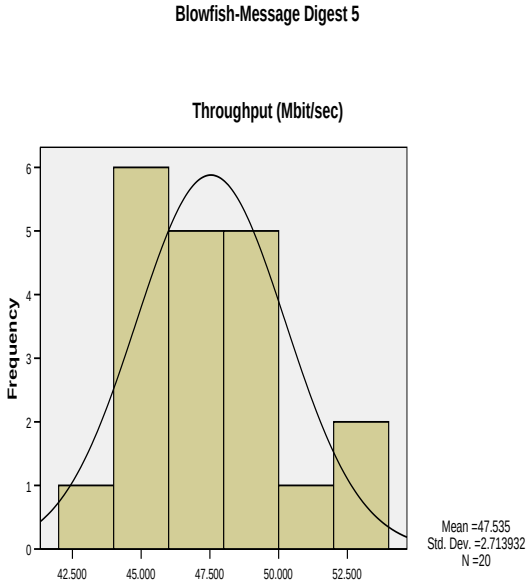

Figure 4.13 BF-MD5 Throughput

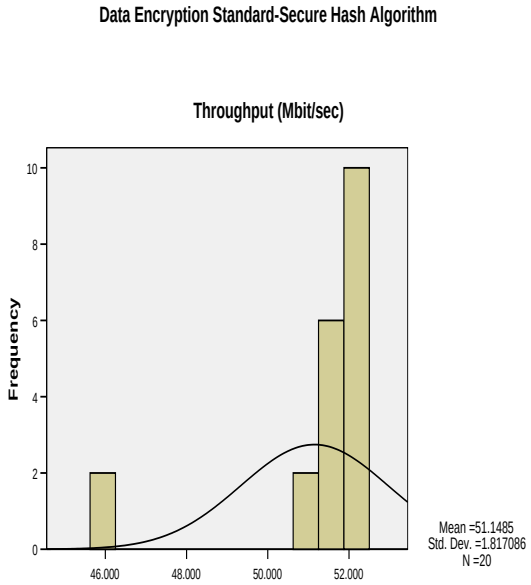

Figure 4.14 DES-SHA Throughput

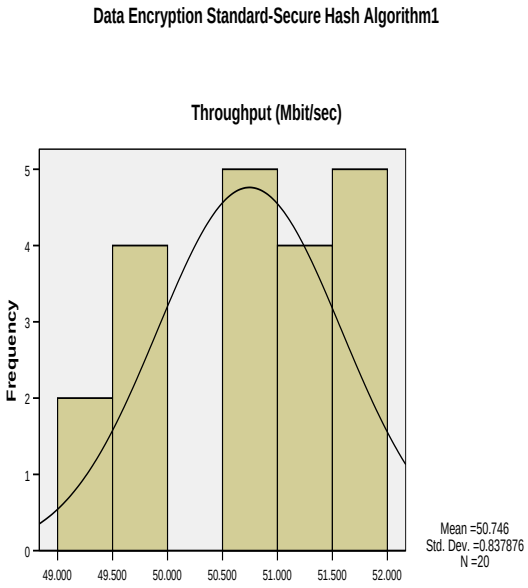

Figure 4.15 DES-SHA1 Throughput

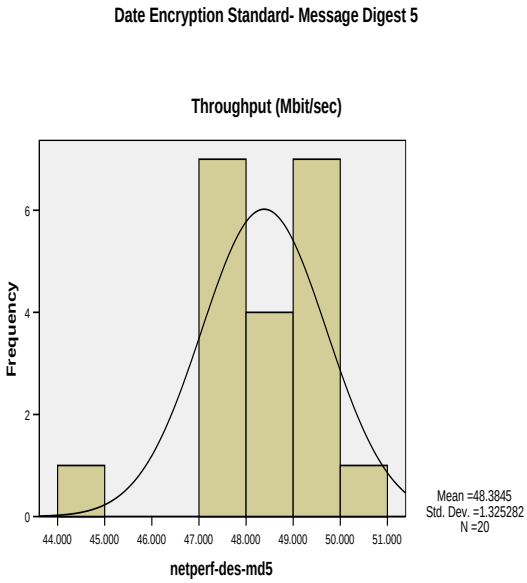

Figure 4.16 DES-MD5 Throughput

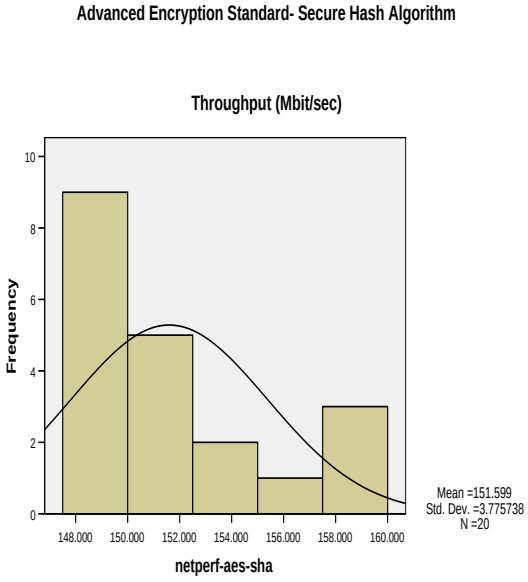

Figure 4.17 AES-SHA Throughput

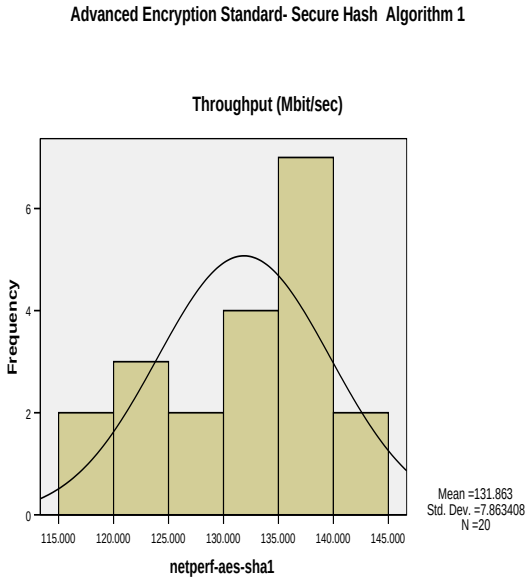

Figure 4.18 AES-SHA1 Throughput

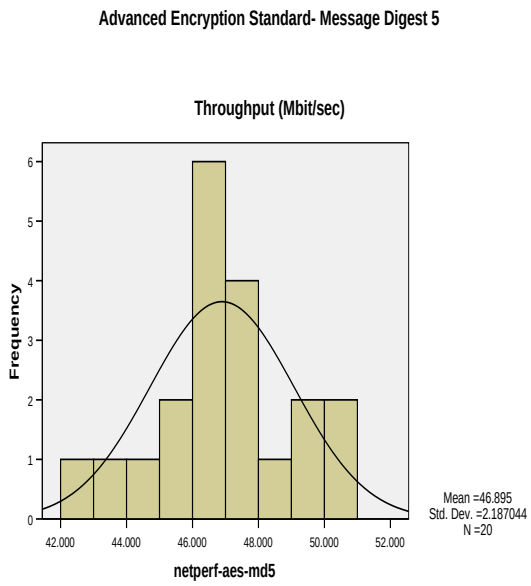

Figure 4.19 AES-MD5 Throughput

Table 4.2 Throughput Statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
netperf-bf-sha	49.439	51.940	35.460	3.891	15.142
netperf-bf-sha1	49.317	51.680	41.680	2.680	7.181
netperf-bf-md5	47.535	52.820	42.910	2.714	7.365
netperf-des-sha	51.149	52.080	45.890	1.817	3.302
netperf-des-sha1	50.746	51.880	49.140	.838	.702
netperf-des-md5	48.385	50.040	44.930	1.325	1.756
netperf-aes-sha	151.599	158.960	147.540	3.776	14.256
netperf-aes-sha1	131.863	144.120	117.380	7.863	61.833
netperf-aes-md5	46.895	50.290	42.080	2.187	4.783

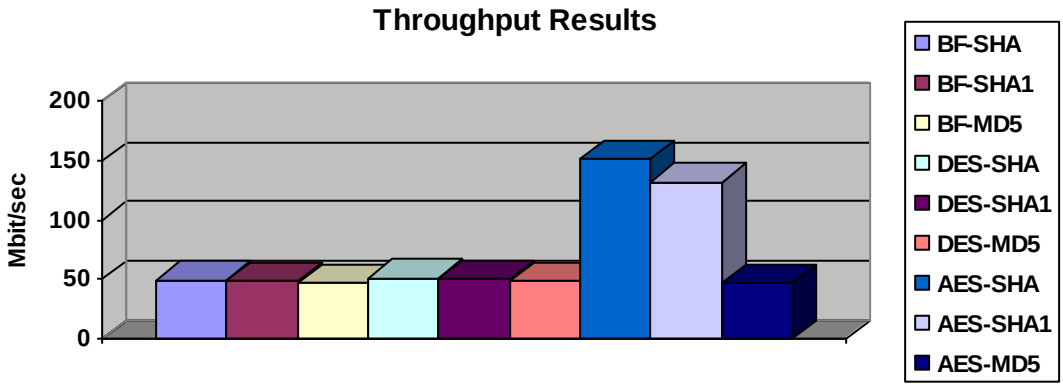

Figure 4.20 Throughput Mean Values

from above figures and table the throughput is not significantly change through all the algorithms except at AES-SHA and AES-SHA1 values, which large difference acquire but the real reason mentioned in section 3.2.4, that this values was taken after change in net service from 1.5 Mbit/s to 5 Mbit/s with fiber optic cable, and the location of the implementation is changed to another lab use switch instead of hub used in previous lab.

So the Throughput doesn't have significant change with encryption and Hash algorithms because the short distance between tow computers (rate effected by transmission time)

4.3.3 TTCP Throughput Results

Below results of throughput (Mbit\sec) for different cipher algorithms and hash functions algorithms

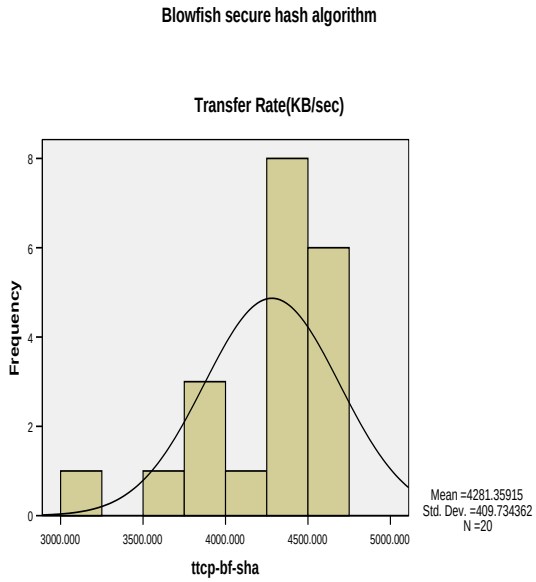

Figure 4.21 BF-SHA Throughput(TTCP)

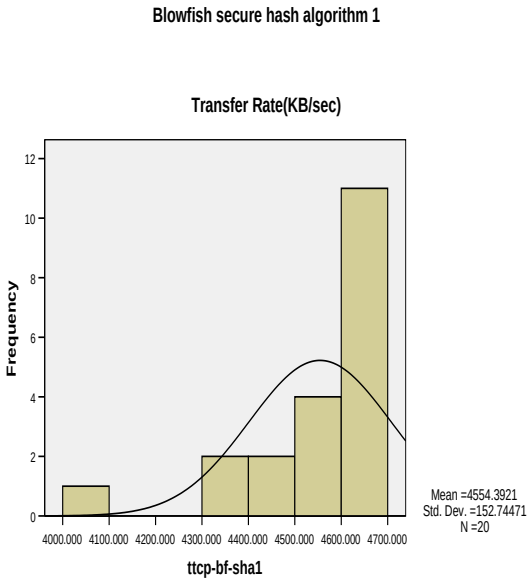

Figure 4.22 BF-SHA1 Throughput(TTCP)

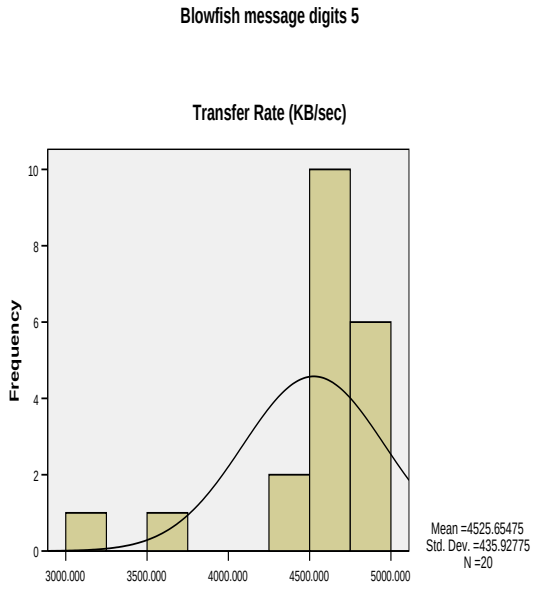

Figure 4.23 BF-MD5 Throughput(TTCP)

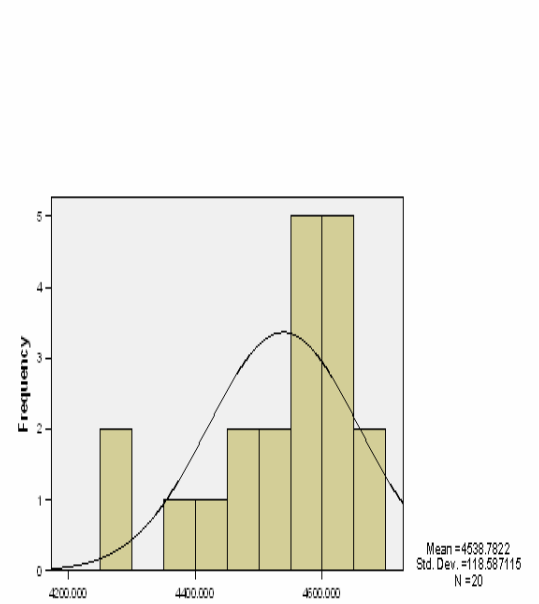

Figure 4.24 DES-SHA Throughput(TTCP)

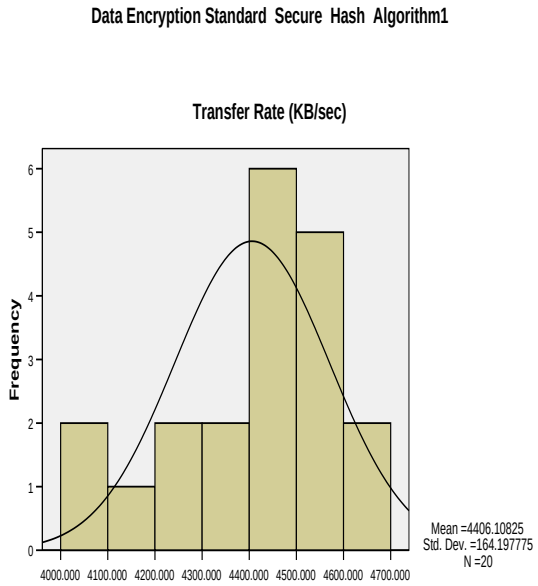

Figure 4.25 DES-SHA1 Throughput(TTCP)

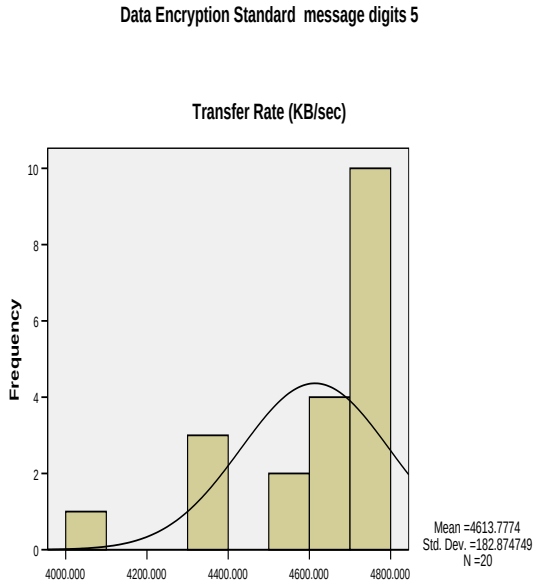

Figure 4.26 DES-MD5 Throughput(TTCP)

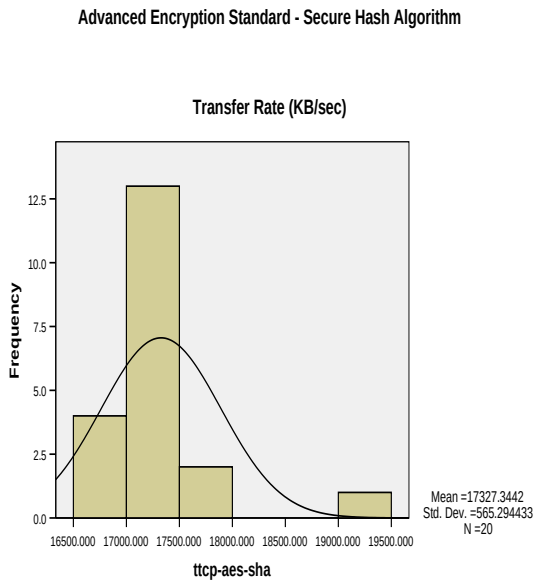

Figure 4.27 AES-SHA Throughput(TTCP)

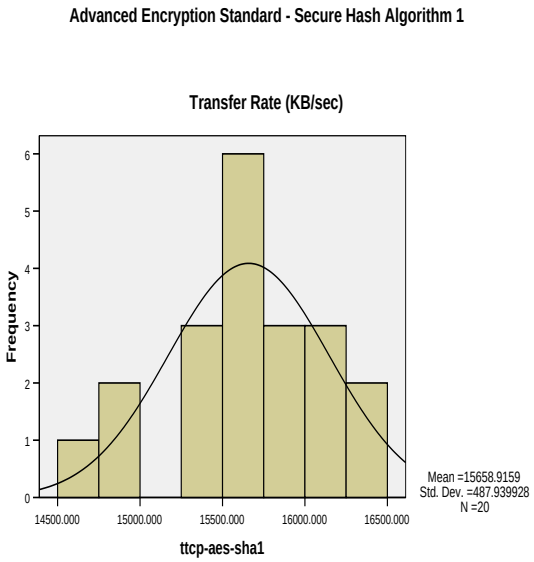

Figure 4.28 AES-SHA1 Throughput(TTCP)

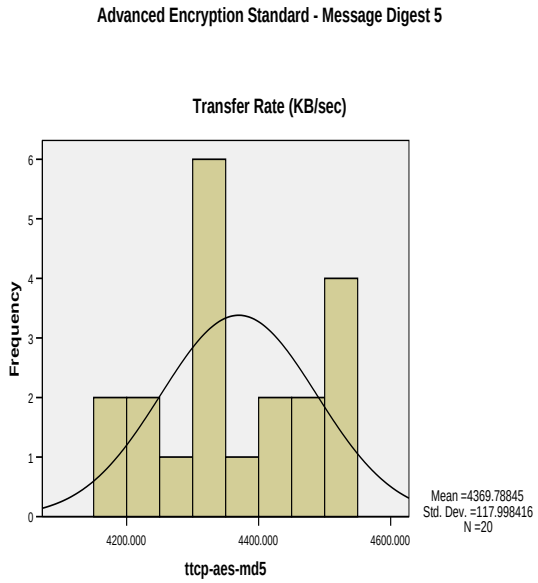

Figure 4.29 BF-SHA Throughput (TTCP)

Table 4.3 Throughput Statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
ttcp-bf-sha	4281.359	4702.862	3123.561	409.734	167882.248
ttcp-bf-sha1	4554.392	4694.546	4098.458	152.745	23330.947
ttcp-bf-md5	4525.655	4785.547	3015.142	435.928	190033.003
ttcp-des-sha	4538.782	4683.719	4285.541	118.587	14062.904
ttcp-des-sha1	4406.108	4609.412	4073.431	164.198	26960.909
ttcp-des-md5	4613.777	4754.964	4065.316	182.875	33443.174
ttcp-aes-sha	17327.344	19498.564	16706.792	565.294	319557.796
ttcp-aes-sha1	15658.916	16320.286	14579.184	487.940	238085.373
ttcp-aes-md5	4369.788	4538.451	4153.674	117.998	13923.626

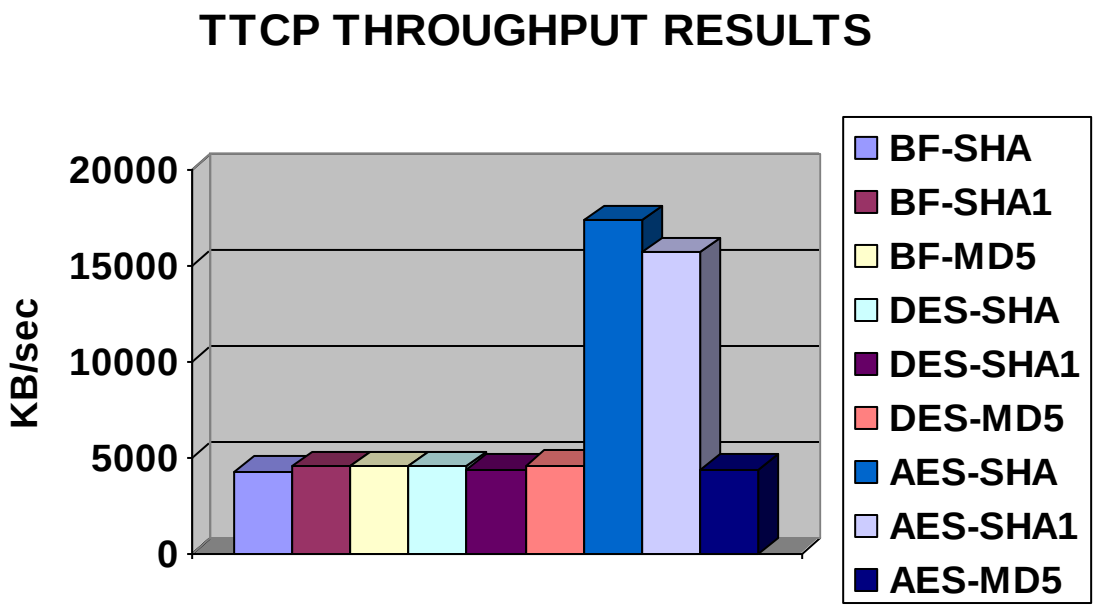

Figure 4.30 Throughput Mean Values Using TTCP

from above figures and table the throughput is not significantly change through all the algorithms except at AES-SHA and AES-SHA1 values, which large difference acquire but the real reason mentioned in section 3.2.4, that this values was taken after change in net service from 1.5 Mbit/s to 5 Mbit/s with fiber optic cable, and the location of the implementation is changed to another lab use switch instead of hub used in previous lab.

So the Throughput doesn't have significant change (there is change but too small) with encryption and Hash algorithms because the short distance between tow computers (rate effected by transmission time).

4.3.4 Ping RTT Results

Below results of RTT (milliseconds) for different cipher algorithms and hash functions algorithms.

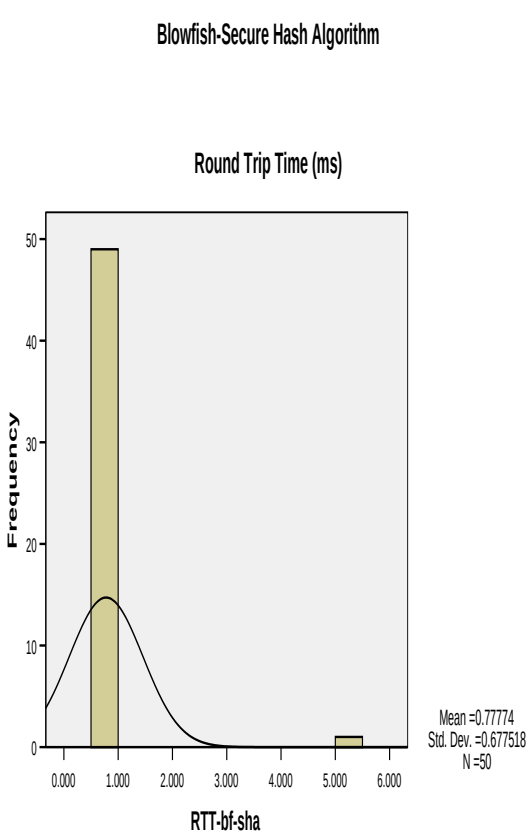

Figure 4.31 BF-SHA RTT

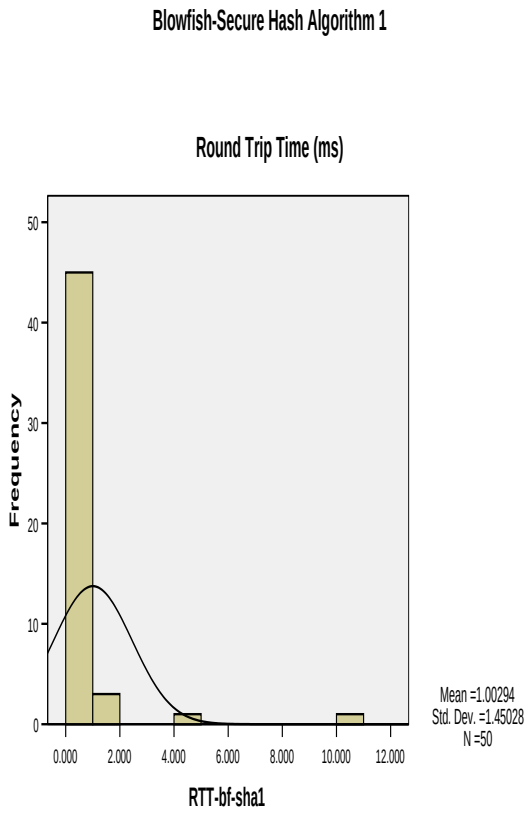

Figure 4.32 BF-SHA1 RTT

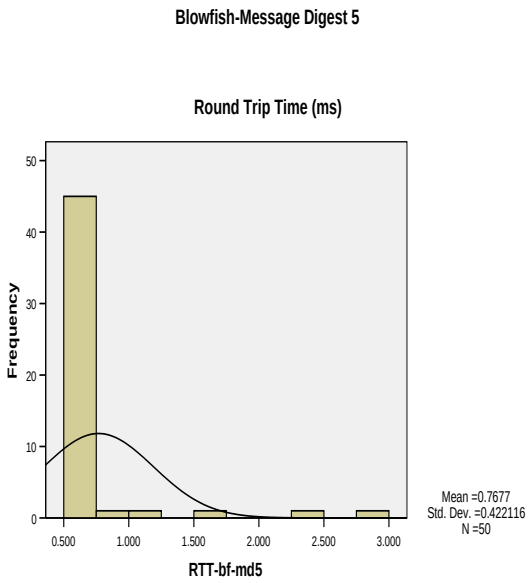

Figure 4.33 BF-MD5 RTT

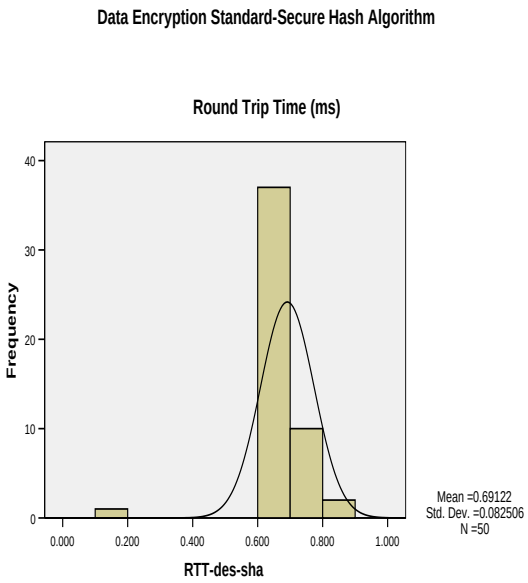

Figure 4.34 DES-SHA RTT

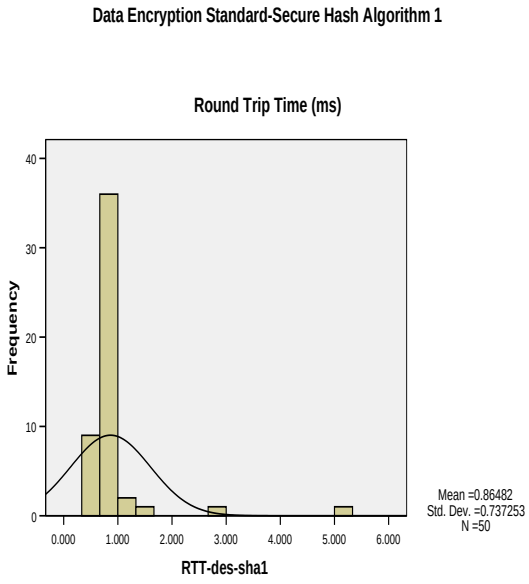

Figure 4.35 DES-SHA1 RTT

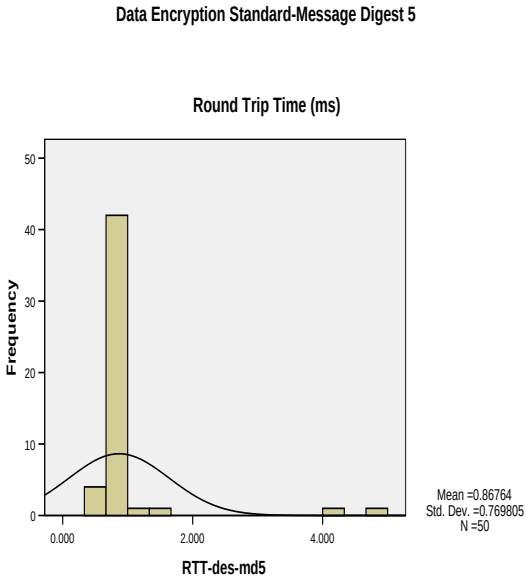

Figure 4.36 DES-MD5 RTT

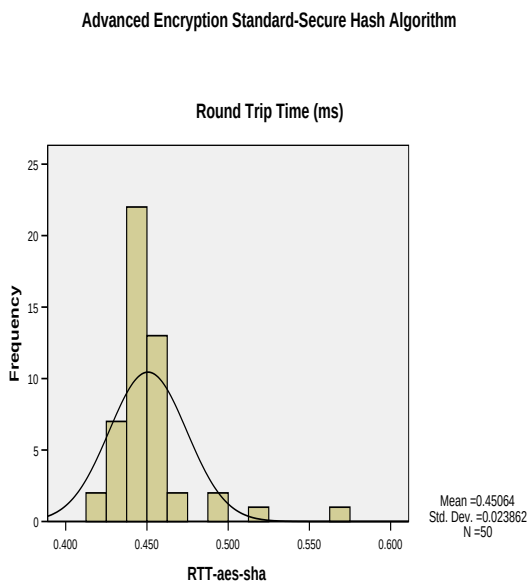

Figure 4.37 AES-SHA RTT

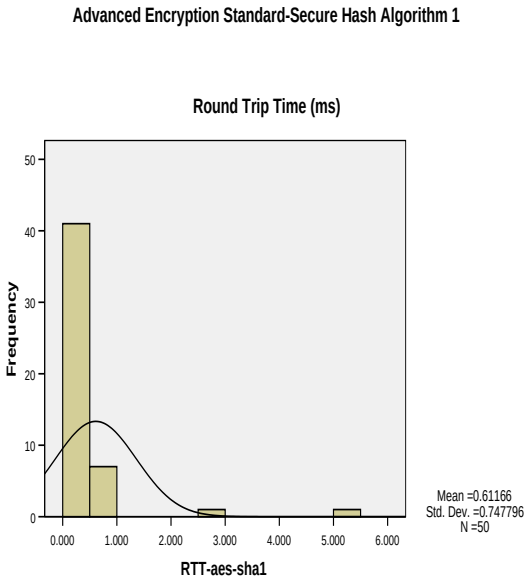

Figure 4.38 AES-SHA1 RTT

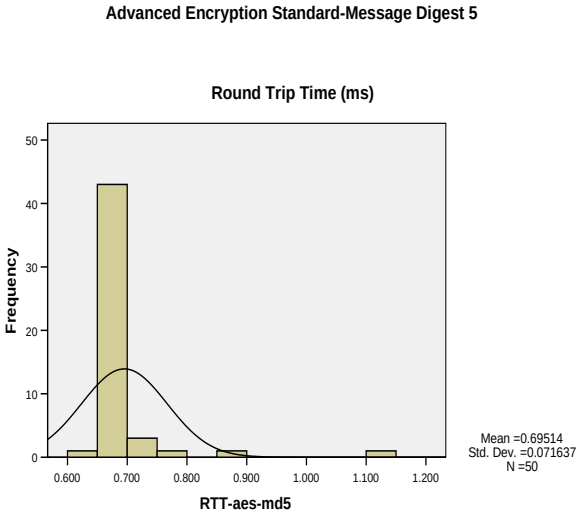

Figure 4.31 AES-MD5 RTT

TABLE 4.4 RTT Statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
RTT-bf-sha	.778	5.470	.631	.678	.459
RTT-bf-sha1	1.003	10.100	.637	1.450	2.103
RTT-bf-md5	.768	2.950	.610	.422	.178
RTT-des-sha	.691	.866	.176	.083	.007
RTT-des-sha1	.865	5.320	.649	.737	.544
RTT-des-md5	.868	4.900	.634	.770	.593
RTT-aes-sha	.451	.567	.416	.024	.001
RTT-aes-sha1	.612	5.360	.428	.748	.559
RTT-aes-md5	.695	1.130	.646	.072	.005

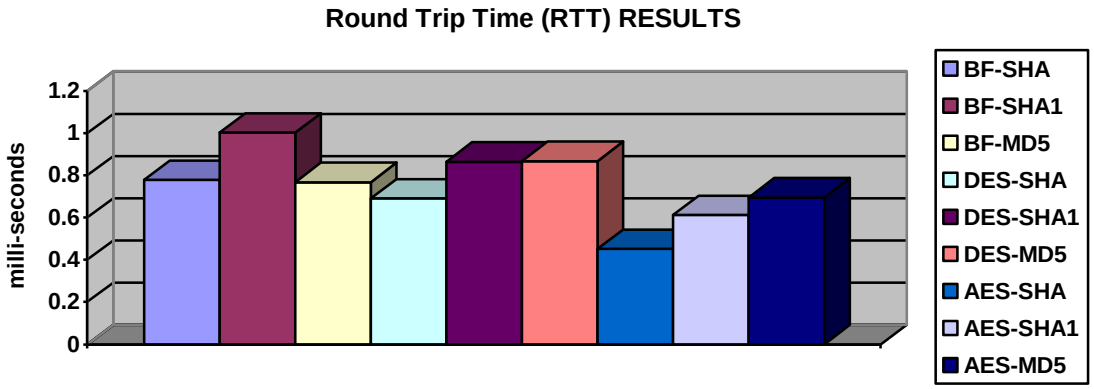

Figure 4.30 RTT Mean Values

So the RTT doesn't have significant change (there is change but too small) with encryption and Hash algorithms because the short distance between tow computers (RTT effected by transmission time).

4.3.5 Iperf Jitter Results:

Below results of Jitter (milliseconds) for different cipher algorithms and hash functions algorithms.

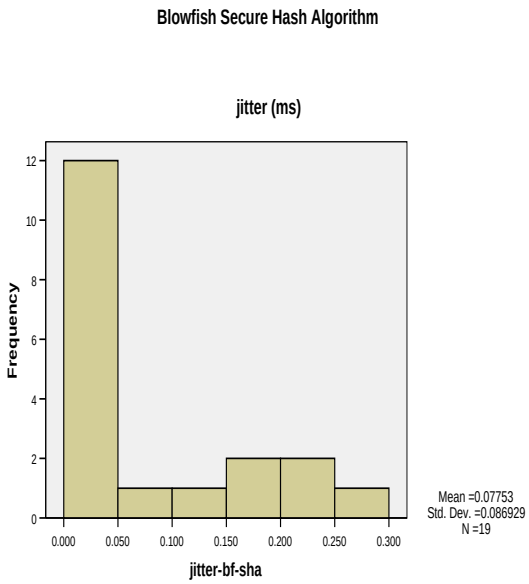

Figure 4.41 BF-SHA Jitter

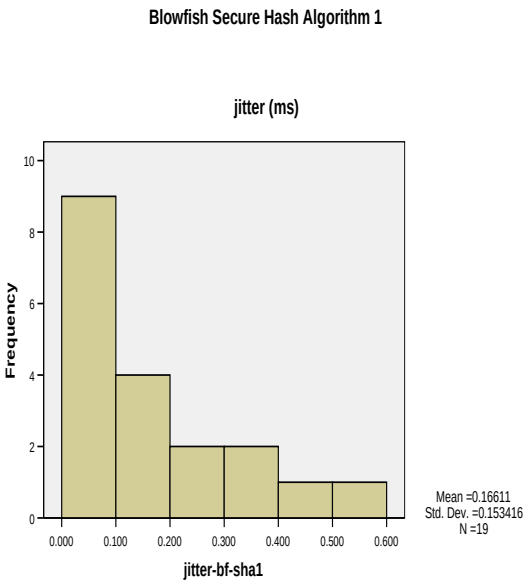

Figure 4.42 BF-SHA1 Jitter

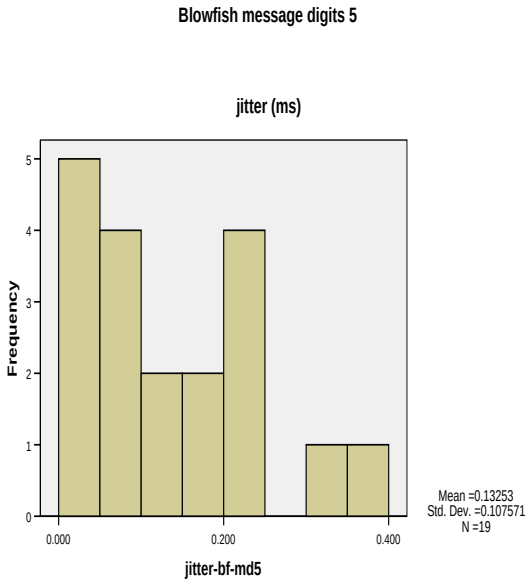

Figure 4.43 BF-MD5 Jitter

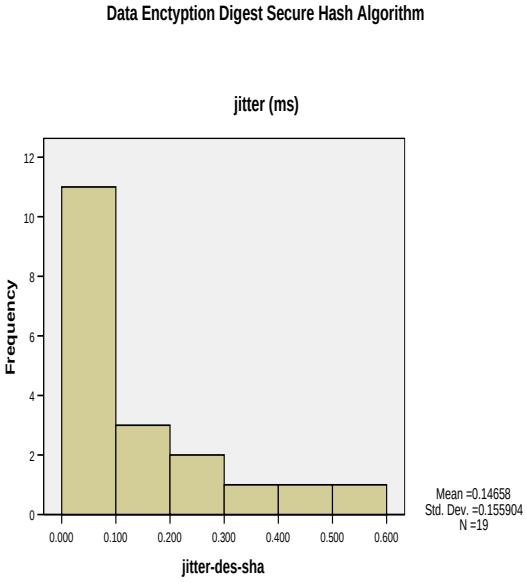

Figure 4.44 DES-SHA Jitter

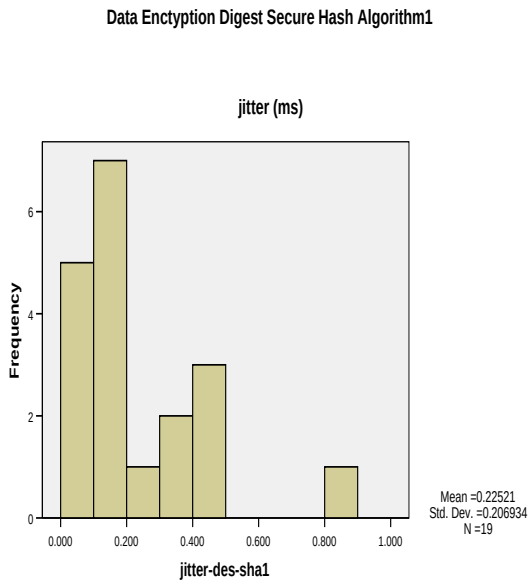

Figure 4.45 DES-SHA1 Jitter

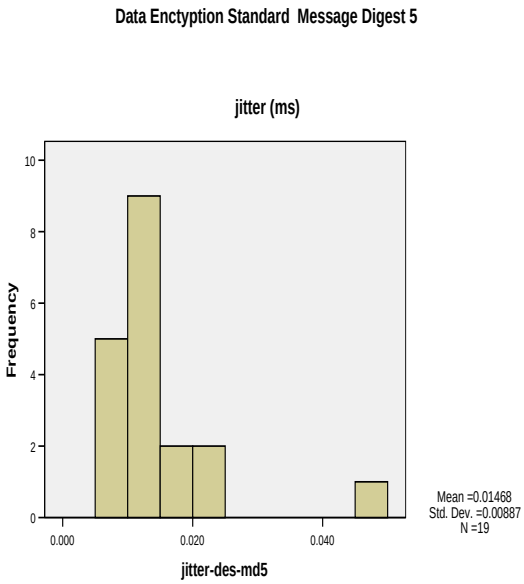

Figure 4.46 DES-MD5 Jitter

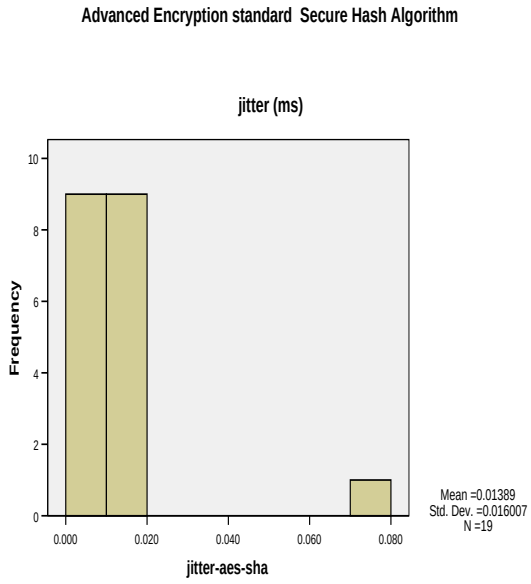

Figure 4.47 AES-SHA Jitter

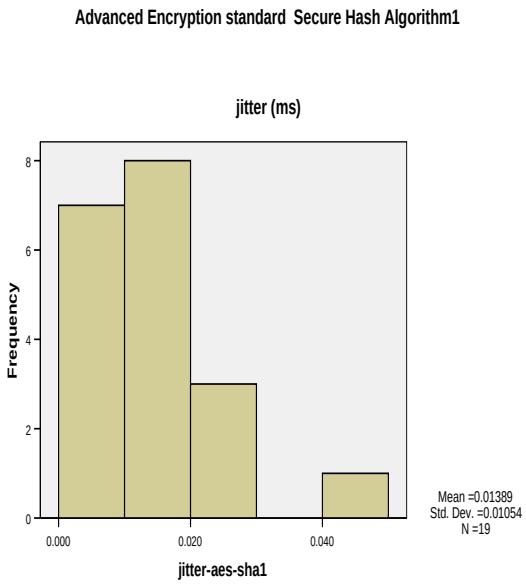

Figure 4.48 AES-SHA1 Jitter

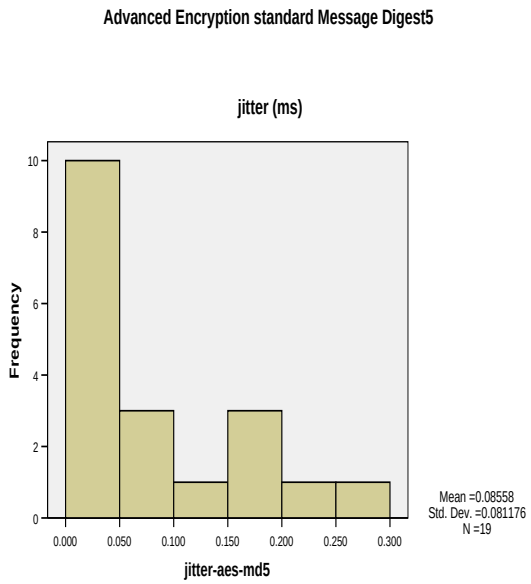

Figure 4.49 AES-MD5 Jitter

Table 4.5 Jitter Statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
jitter-bf-sha	.078	.268	.011	.087	.00756
jitter-bf-sha1	.166	.533	.010	.153	.02354
jitter-bf-md5	.133	.354	.009	.108	.01157
jitter-des-sha	.147	.572	.013	.156	.02431
jitter-des-sha1	.225	.832	.028	.207	.04282
jitter-des-md5	.015	.047	.007	.009	.00008
jitter-aes-sha	.014	.079	.006	.016	.00026
jitter-aes-sha1	.014	.049	.003	.011	.00011
jitter-aes-md5	.086	.283	.010	.081	.00659

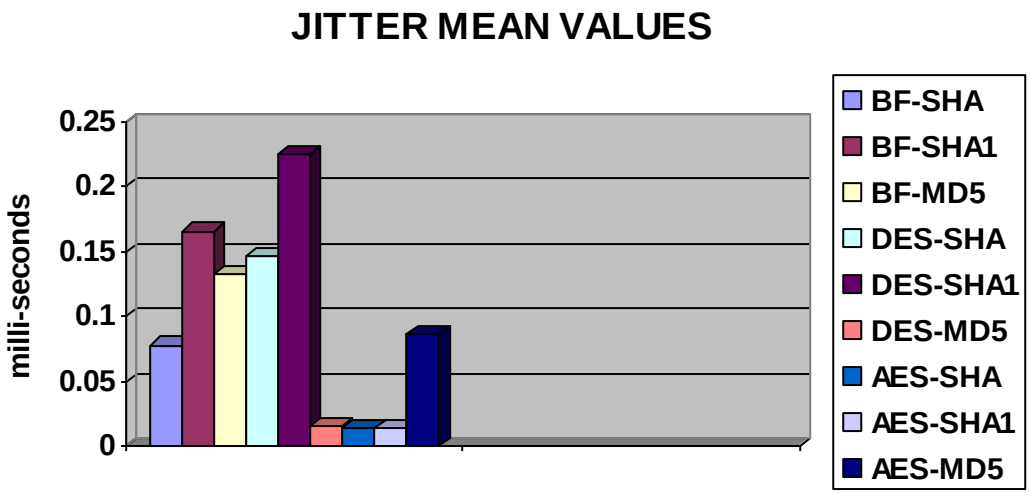

Figure 4.50 Jitter Mean Values

From figures(4.41 to 4.49), the result concentrated on 0 to 0.02 ms but there are some large values at about .5 ms, so the mean doesn't give good directions , in general the Jitter's change doesn't depend of algorithms. This type of jitter called random jitter

4.3.6 Ping Packet Loss Results

Below results of Packet loss for different cipher algorithms and hash functions algorithms , there is no Packet loss during all experiments.

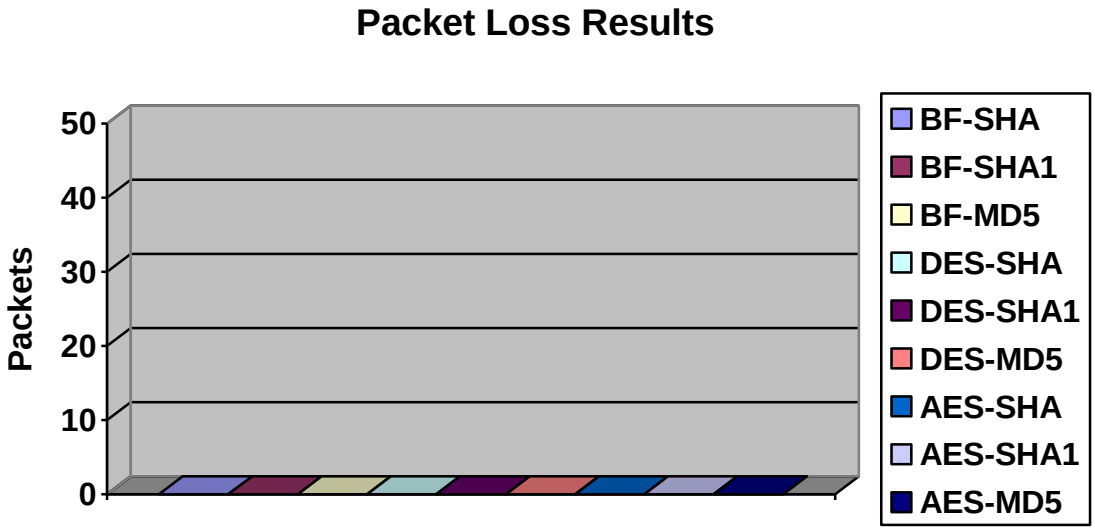

Figure 4.51 Packet Loss

4.3.7 FTP Transfer Rate Results

Below results of a transfer rate of FTP (KB\sec) for different cipher algorithms and hash functions algorithms

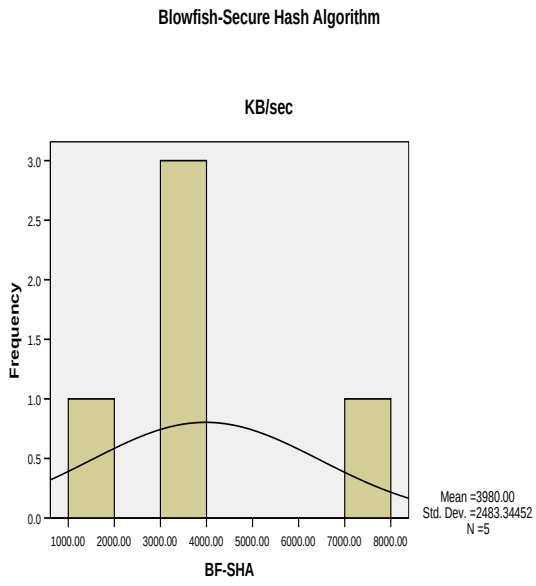

Figure 4.52 BF-SHA FTP Transfer Rate

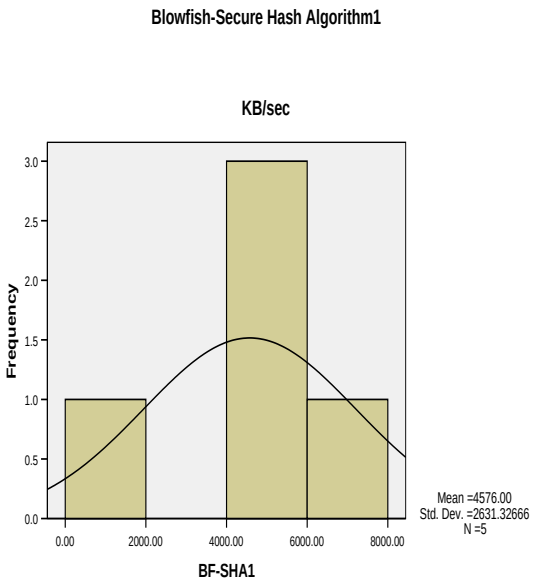

Figure 4.53 BF-SHA1 FTP Transfer Rate

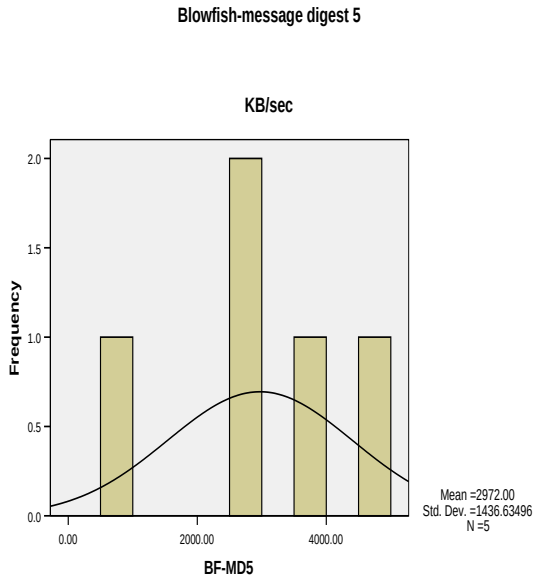

Figure 4.54 BF-MD5 FTP Transfer Rate

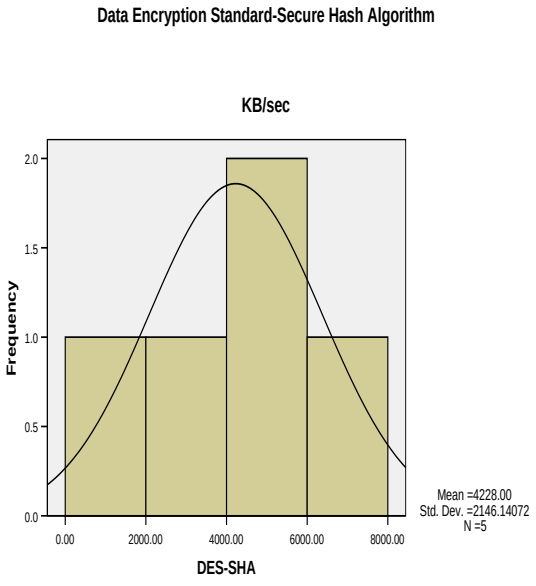

Figure 4.55 DES-SHA FTP Transfer Rate

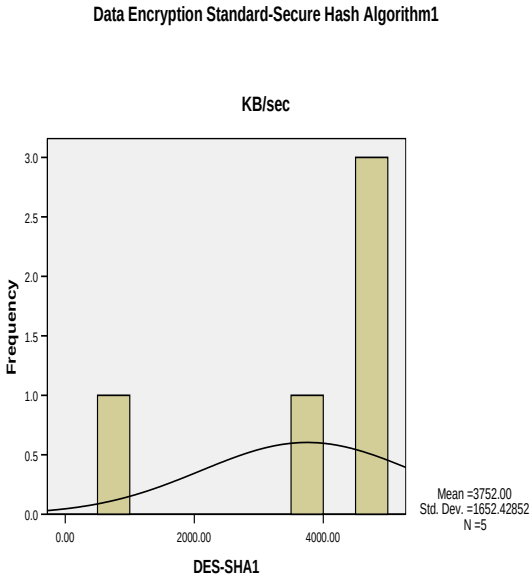

Figure 4.56 DES-SHA1 FTP Transfer Rate

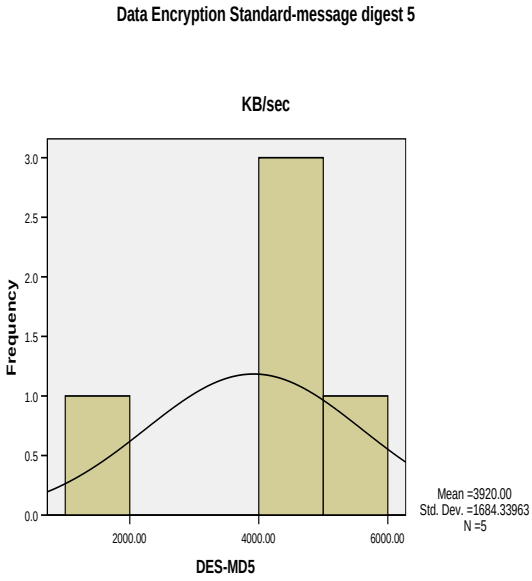

Figure 4.57 DES-MD5 FTP Transfer Rate

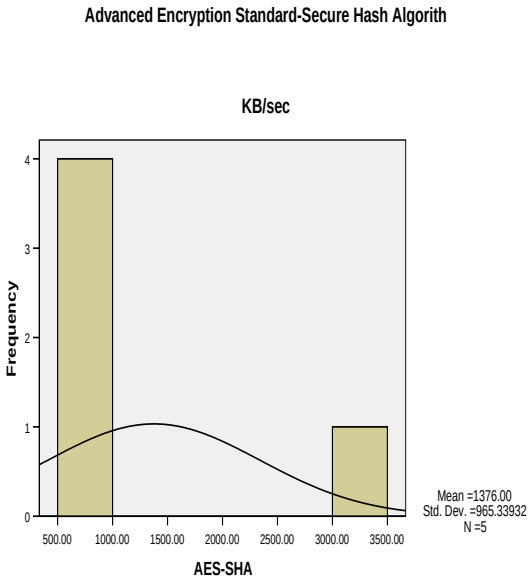

Figure 4.58 AES-SHA FTP Transfer Rate

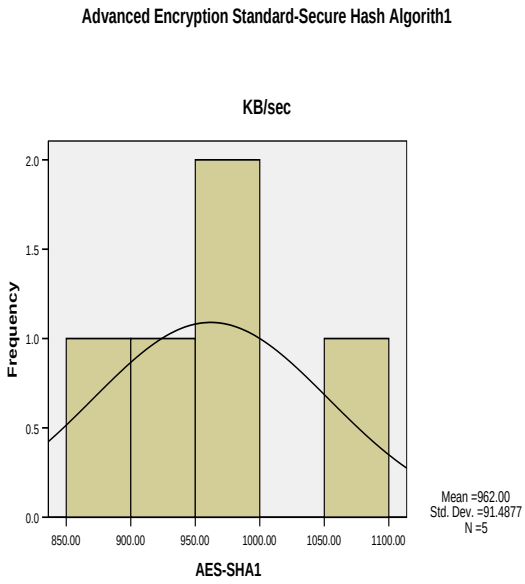

Figure 4.59 AES-SHA1 FTP Transfer Rate

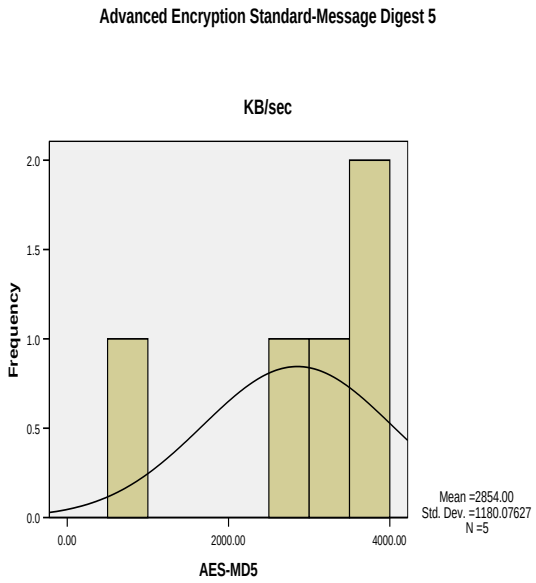

Figure 4.60 AES-MD5 FTP Transfer Rate

TABLE 4.6 FTP Transfer Rate Statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
Bf-sha	3980.00	7900.00	1000.00	2483.34	6167000.00
Bf-sha1	4576.00	8000.00	880.00	2631.33	6923880.00
Bf-md5	2972.00	4800.00	860.00	1436.63	2063920.00
Des-sha	4228.00	6100.00	940.00	2146.14	4605920.00
Des-sha1	3752.00	4800.00	860.00	1652.43	2730520.00
Des-md5	3920.00	5100.00	1000.00	1684.34	2837000.00
Aes-sha	1376.00	3100.00	850.00	965.34	931880.00
Aes-sha1	962.00	1100.00	850.00	91.49	8370.00
Aes-md5	2854.00	3800.00	870.00	1180.08	1392580.00

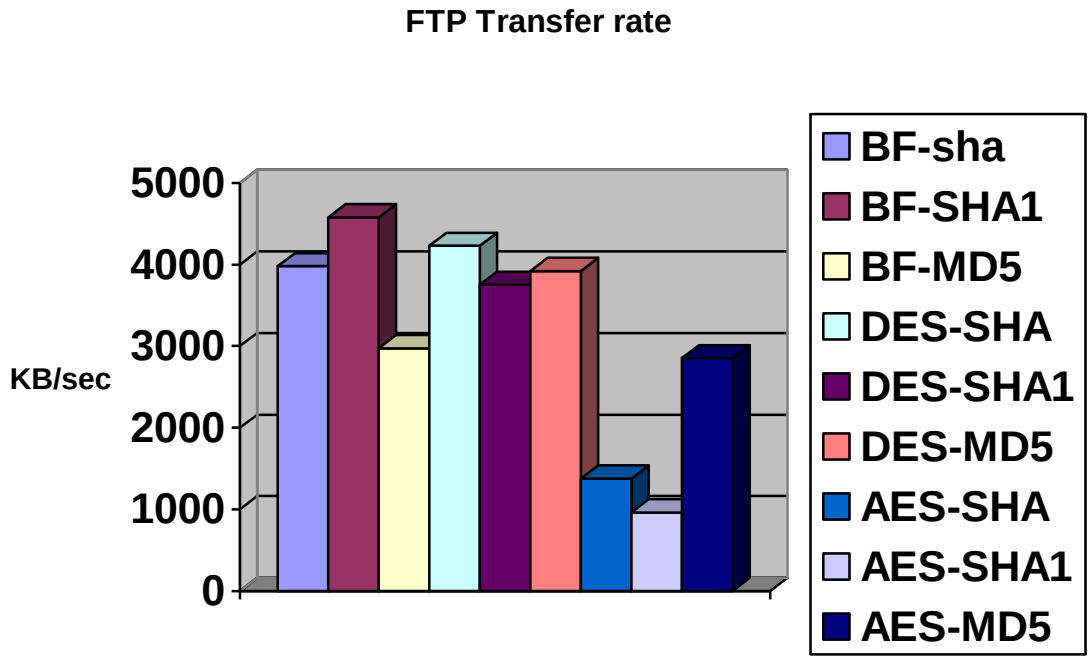

Figure 4.61 FTP Transfer Rate Mean Values

From above figure, the transfer rate is less in AES than other algorithms; we can conclude that the transfer rate is reduced in large keys.

4.2.8 NFS Access time Results

Below result for NFS access time measured using TCPDUMP Application

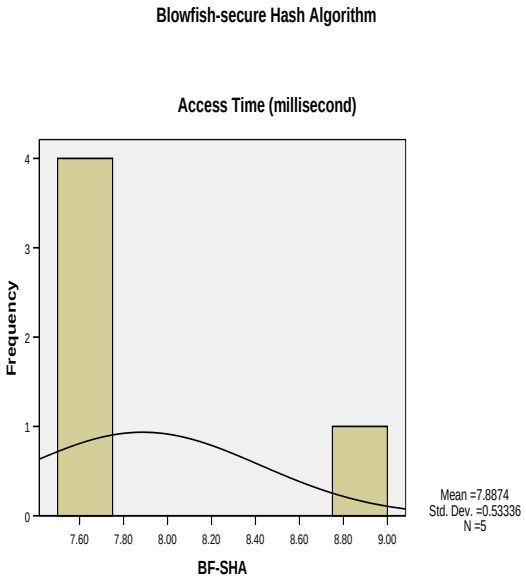

Figure 4.62 BF-SHA NFS Access Time

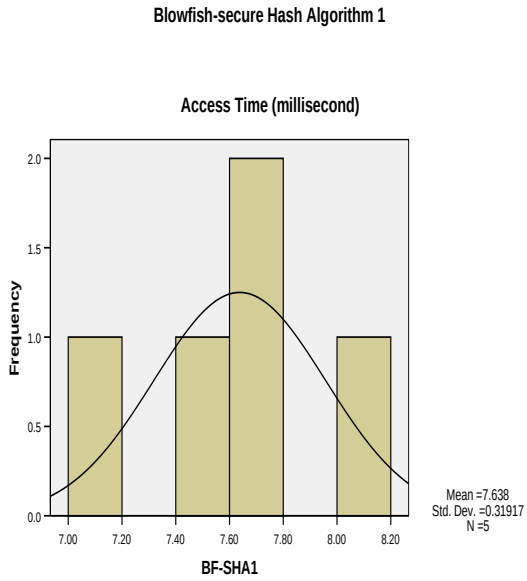

Figure 4.63 BF-SHA FTP Access Time

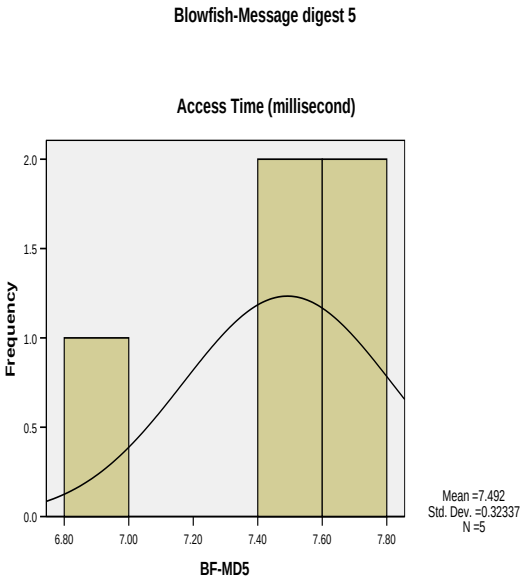

Figure 4.64 BF-MD5 NFS Access Time

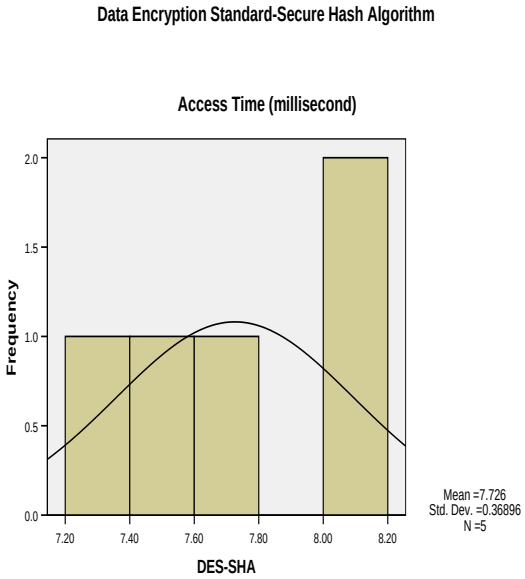

Figure 4.65 DES-SHA NFS Access Time

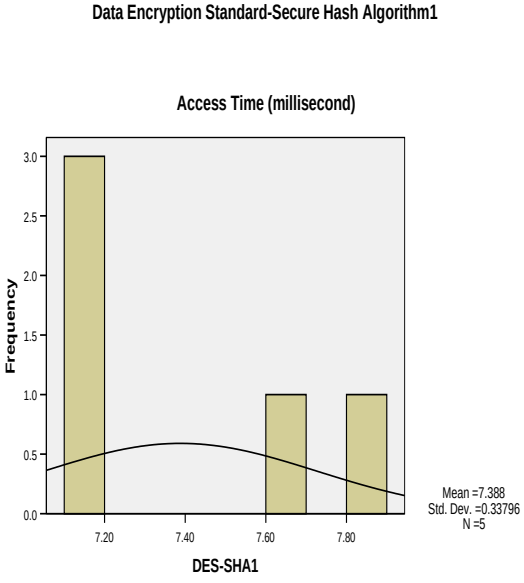

Figure 4.66 DES-SHA1 NFS Access Time

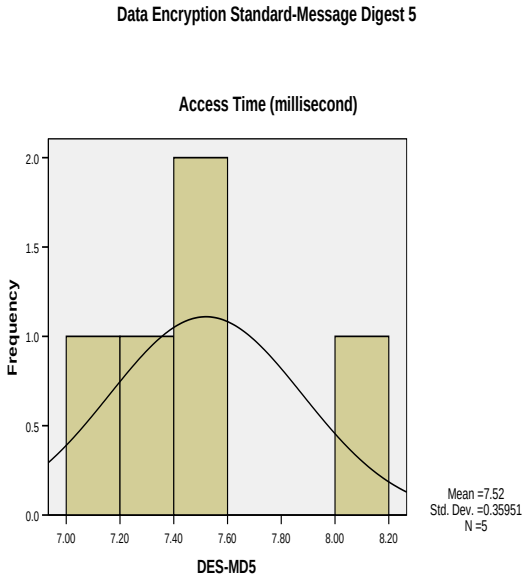

Figure 4.67 DES-MD5 NFS Access Time

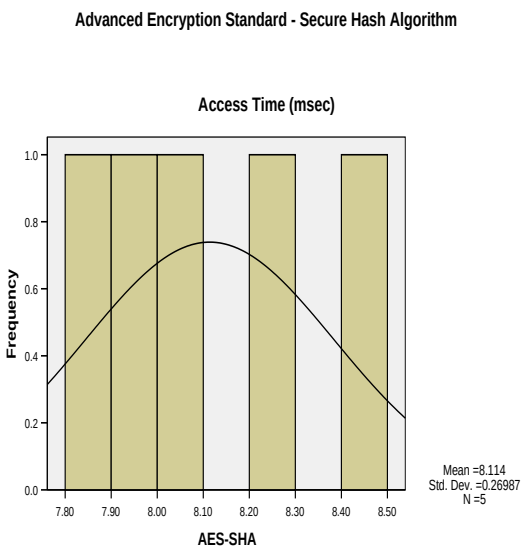

Figure 4.68 AES-SHA NFS Access Time

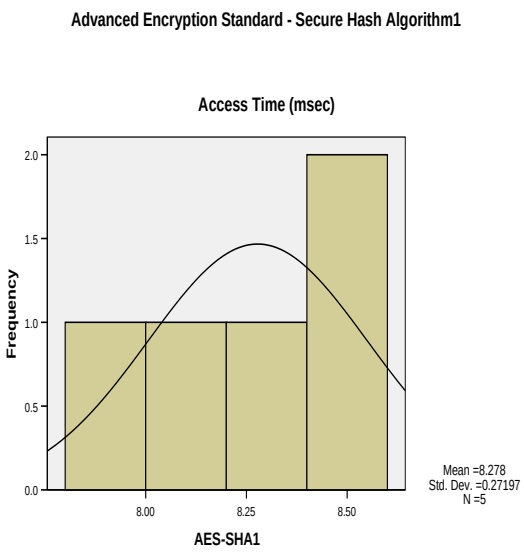

Figure 4.69 AES-SHA1 NFS Access Time

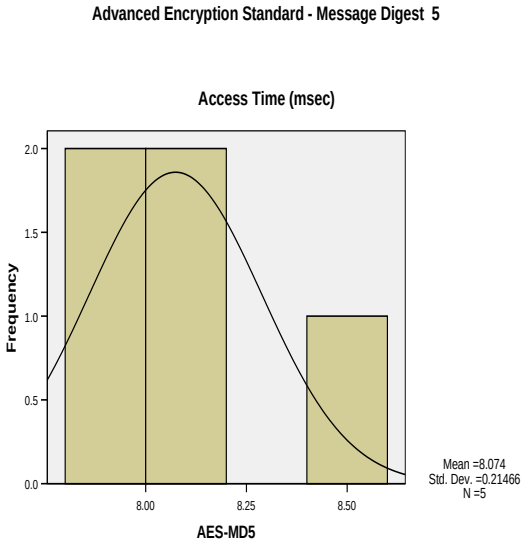

Figure 4.70 AES-MD5 NFS Access Time

Table 4.7 NFS Transfer Rate statistics

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
BF-SHA	7.89	8.84	7.57	.53	.28
BF-SHA1	7.64	8.03	7.16	.32	.10
BF-MD5	7.49	7.80	6.95	.32	.10
DES-SHA	7.73	8.14	7.24	.37	.14
DES-SHA1	7.39	7.85	7.14	.34	.11
DES-MD5	7.52	8.12	7.18	.36	.13
AES-SHA	8.11	8.49	7.82	.27	.07
AES-SHA1	8.28	8.60	7.89	.27	.07
AES-MD5	8.07	8.42	7.87	.21	.05

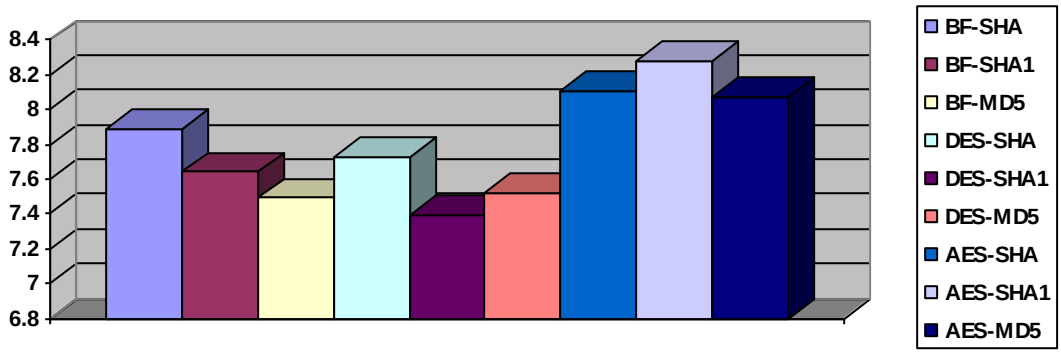

Figure 4.71 NFS Transfer Rate Means Values

From above figure, AES access time is the highest one, we can get that the mounting access time is increases (performance was degraded) with large key (256). This result appropriate with FTP transfer rate.

4.4 IPsec Results

This section summarizes tests result for work hardware IPsec VPN network we measure RTT using Pinging and Throughput using Jperf software. (Tests was taken from IPSEC VPN in MTN)

TABLE 4.8 IPsec VPN RTT and Throughput

Test	Mean	Maximum	Minimum	Standard Deviation	Variance
Round Trip Time	7.24	8.10	5.94	.50	.25
Iperf (Throughput)	231.88	242.47	217.90	9.67	93.42

4.4.1 RTT Result

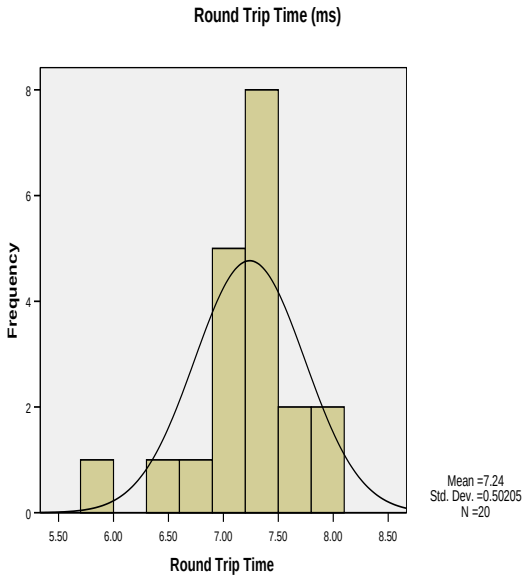

Figure 4.72 IPsec RTT

4.4.2 Throughput Result

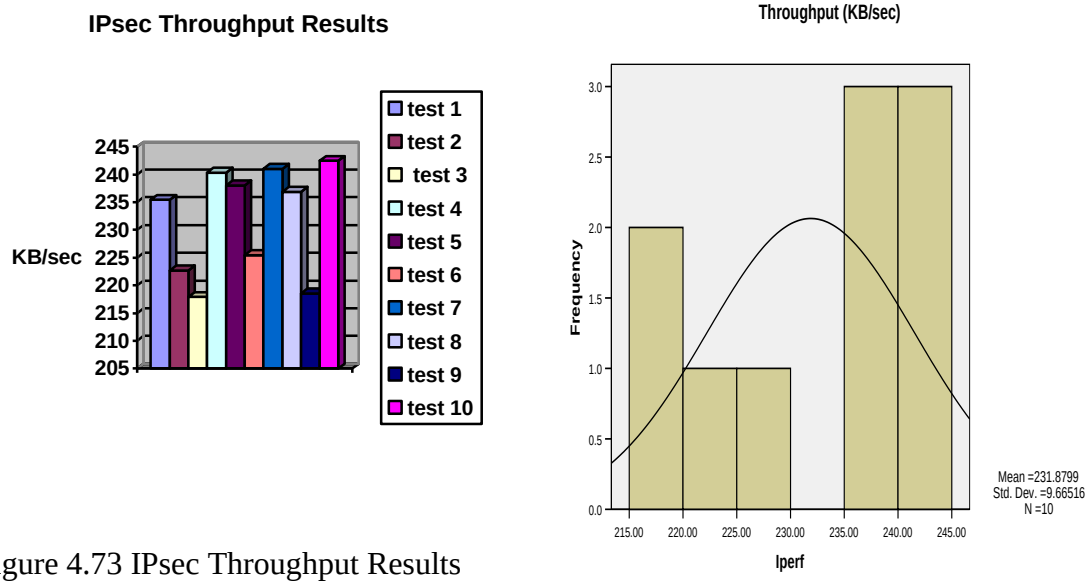

Figure 4.73 IPsec Throughput Results

From the above table and figure the RTT time is more than 5 ms which is too larger than SSL VPN value which is less than 1 ms

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

In this project the SSL VPN has successfully been designed and implemented and the effects of changing cryptography algorithms on the VPN performance has been discussed, also we take readings of IPsec VPN performance from one of MTN networks. The SSL VPN design using Openvpn software is sufficiently accomplished the Design goals specified in Chapter 3 we can summarize achieved goals below:

- the SSL VPN design give user authentication , where Openvpn give each user private secrete key, which will be generated in server device and moved to the client device using flash memory or floppy desk, also provide message authentication using HASH functions algorithms .
- The most important required characteristic in VPN design that the tunneling protocol (SSL) must provide enough security to the data, this goal achieved using encryption algorithms.
- The design also supported with some application protocols (FTP, NFS) to simplify the performance evaluation of the network, such as measuring file transfer rate.
- The SSL VPN monitored using TCPDUMP software, which monitor the packets and acknowledges transferred between client and server related to the time. This monitoring help in measuring access time of NFS mounting.

As mentioned the performance of network is evaluated using some software, the measures is include transfer rate, bandwidth, jitter, RTT and packet loss, the performance lead to main conclusions :

The performance of network is become slower when data encrypted for example RTT is increase to the double of abstract data, since the ping between any two

Computer in the LAN takes about 0.3 ms and the ping through VPN about 0.7ms.

- The performance of network is affected by key size of encryption algorithms, the performance degraded with large key size, so the security using large size is improved since the performance is degraded, but not all network specification effected by changing of encryption algorithm (key size).

- Performance of SSL VPN is comparable with IPsec VPN and may better in equal environment of implementation. But we can not make direct comparison because of the difference in circumstances.

5.2 Limitations:

- The best choice of implementing of SSL VPN is the complete hardware implementation using VPN appliance, the major limitations in hardware implementation is the high cost and charging.

- One of the most important problems faced the performance evaluation of our design is the instability in net service in university which changed during the testing; this problem is discussed in section 3.2.4.2 and appear at the performance of the results.

- One limitation in the design, it is not client less system but this shortage hasn't effects on the performance evaluation of SSL VPN.

- Another limitation in the design is that our VPN not contain user interface, so the user must be familiar with Linux shell.

5.3 Recommendation for Future work:

- The design can be improved by adding user interface, since Openvpn open source hasn't Linux user interface.

- As mentioned the best choice of VPN implementation is hardware one so future work could include comparing these results to a Cisco-based VPN solution to see If performance is improved in a proprietary, hardware-based implementation.

- The ability to improve the security of the VPN without degrading the performance of the VPN will be good field of research, because the VPN will be too important in the future. May be by improving the security using any technique without increase the size of the key.

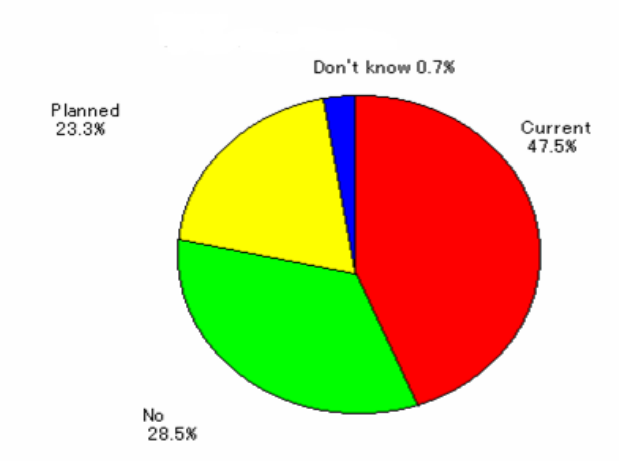

Figure 5.1 VPN in US Companies [13]

References

[1] Howstuffwork, *How VPN Work*, URL: Http:// howstuffwork.com, accessed on April 2008.

[2] Behrouz A. Forouzan, *TCP/IP Protocol Suite*, Third Edition, Tata-nc grow Hill, 2006.

[3] Microsoft, *Virtual Private Networking: An overview*, A White Paper, 2006.

[4]Joseph Steibeng, and Timothy Speed, *SSL VPN*, First Edition, *Packet* Publishing, 2005.

[5] Openmaniak, *What is The Openvpn*, URL: http://openmaniak.com/openvpn.php, accessed at April 2008.

[6] Charlie Hosner, *OpenVPN and SSL VPN Revolution*, Second Edition, SANS Institute, 2004.

[7] *CryptoStuff* Cryptography Tutorial, *HMAC Algorithm in Detail,* URL: http://www.cryptostuff.com/crypto, accessed at June 2008.

[8] Wikipedia, *Public Key Infrastructure*, URL: http://en.wikipedia.org/wiki/Public_key_infrastructure, accessed on April 2008.

[9] Terry Collings, and Kurt Wall, *Red Hat Linux Networking and System Administration*, Third Edition, Wiley, 2006.

[10] SearchUnifiedCommunications.com Definitions, Jitter, URL: http://searchunifiedcommunications.techtarget.com/sDefinition/0,,sid186_gci213534,00.html accessed on May 2008.

[11] Rajesh and Sajesh, *Openvpn Installation and Configuration*, 2006, URL: www.esnips.com/_t_/openvpn , accessed on February 2008.

[12] Joseph D. Sloan, *Network Troubleshooting Tools*, Second Edition, O'Reill*y*, 2001

[13] Germaine Bacon, Lizzi Beduya, Betty Huang, Jun Mitsuoka, and Juliet Polintan *Virtual Private Network,* URL: http://uniforum.chi.il.us, accessed on June 2008.

Appendix A

This appendix is introduced to give more details to Openvpn user About Openvpn Configuration .

1. Install OpenVPN package on your system (server and client). After that you can use below command to ensure

rpm -qa ‘‘*openvpn*’’

2. Copy the configuration files from /usr/share/doc/openvpn to /etc/openvpn

cp /usr/share/doc/openvpn/easy-rsa/2.0/* /etc/openvpn

3. Now edit the **vars** file and set the KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG, and KEY_EMAIL parameters. Don't leave any of these parameters blank.

Cd /etc/openvpn

vi vars /* editing */

4. Initialize the *PKI*.

#. /vars

#. /clean-all

#. /build-ca

5. Check to see the keys got created. They should be in a subdirectory called **keys**.

ls keys

6. Next, build the *server key*.

#. /build-key-server server

7. Generate *certificates* & *keys* for one client

#. /build-key client1

8. Generate *Diffie Hellman* parameters

#. /build-dh

9. Copy the **server.conf** file from the **/usr/share/doc/openvpn-2.1/sample-config-files** directory to **/etc/openvpn**, and make the suitable changes in options. See appendix A
10. Copy the **client.conf** file to **/etc/openvpn** and make changes to be appropriated to the design environment. See appendix A

And all clients

11. Copy the **client.conf** file and associated keys (**client1.key, client1.crt**) to the client device, and place into the **/etc/openvpn** directory.
12. On server device start the OpenVPN server from the command line rather than start it as a service :

```
# openvpn /etc/openvpn/server.conf
```

13. On client device start the OpenVPN client from the command line rather than start it as a service:

```
# openvpn /etc/openvpn/client.conf
```

14. try a ping across the VPN from the client :

```
# Ping 10.8.0.1
```

The suggested results of Pinging are below:

```
Sun Feb 6 20:46:38 2005 TLS-Auth MTU parms [ L:1542 D:138 EF:38 EB:0 ET:0 EL:0 ]
Sun Feb 6 20:46:38 2005 TUN/TAP device tun1 opened
Sun Feb 6 20:46:38 2005 /sbin/ifconfig tun1 10.8.0.1 pointopoint 10.8.0.2 mtu 1500
Sun Feb 6 20:46:38 2005 /sbin/route add -net 10.8.0.0 netmask 255.255.255.0 gw 10.8.0.2
Sun Feb 6 20:46:38 2005 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:23 ET:0 EL:0 AF:3/1 ]
Sun Feb 6 20:46:38 2005 UDPv4 link local (bound): [undef]:1194
Sun Feb 6 20:46:38 2005 UDPv4 link remote: [undef]
Sun Feb 6 20:46:38 2005 MULTI: multi_init called, r=256 v=256
Sun Feb 6 20:46:38 2005 IFCONFIG POOL: base=10.8.0.4 size=62
Sun Feb 6 20:46:38 2005 IFCONFIG POOL LIST
Sun Feb 6 20:46:38 2005 Initialization Sequence Complete
```

Appendix B:

This appendix is introduced to give the Openvpn server's and Openvpn client's configuration files that used in the design, and some directives to select right option suitable to the design.

B.1 The server’s configuration file

Which TCP/UDP port should OpenVPN listens on?

you will need to open up this port on your firewall.

port 1194

TCP or UDP server?

; Proto tcp

proto udp

"dev tun" will create a routed IP tunnel,

"dev tap" will create an ethernet tunnel.

;dev tap

dev tun

SSL/TLS root certificate (ca), certificate

(cert), and private key (key). Each client

and the server must have their own cert and

key file. The server and all clients will

use the same ca file.

ca "/etc/openvpn/ca.crt"

cert "/etc/openvpn/uofk.crt"

key "/etc/openvpn/uofk.key" # This file should be kept secret

Diffie hellman parameters.

```
dh "/etc/openvpn/dh1024.pem"
# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1.
server 10.8.0.0 255.255.255.0
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
; cipher BF-CBC        # Blowfish (default)
; cipher AES-256-CBC   # AES
cipher DES-EDE3-CBC  # Triple-DES
# select hash algorithm
; auth SHA
```

; auth SHA1

auth MD5

Enable compression on the VPN link.

If you enable it here, you must also

enable it in the client config file.

comp-lzo

The persist options will try to avoid

accessing certain resources on restart

that may no longer be accessible because

of the privilege downgrade.

persist-key

persist-tun

Output a short status file showing

current connections, truncated

and rewritten every minute.

status openvpn-status.log

Set the appropriate level of log

file verbosity.

0 is silent, except for fatal errors

4 is reasonable for general usage

5 and 6 can help to debug connection problems

9 is extremely verbose

Verb 5

NOTE: If you are doing this under Windows, the file is called server.ovpn, not server.conf

B.2 The Client's configuration file

```
Client

;dev tap

dev tun

;proto tcp

proto udp

remote 172.16.15.106  1194

resolv-retry infinite

nobind

persist-key

persist-tun

ca "/etc/openvpn/ca.crt"

cert "/etc/openvpn/client1.crt"

key "/etc/openvpn/client1.key"

ns-cert-type server

;cipher DES-EDE3-CBC

;cipher BF-CBC

cipher AES-128-CBC

auth SHA1

ifconfig 10.8.0.2 10.8.0.1

ifconfig 10.8.0.2 255.255.255.0

pull

comp-lzo

verb 5
```

NOTE 1: The file is called **client.ovpn** on Windows

NOTE 2: each client generated should have it is own **cert**/**key** pair. Only the **ca** file is universal across the OpenVPN server.