1 Introduction

1.1 Overview

Until very recently enterprises that needed to link computers in geographically dispersed locations had to build their own wide area networks (WANs). WANs are implemented using leased lines which provide a good performance, reliability and security to networks; on the other hand it is expensive to maintain WANs that are built using leased lines and cost increases with the distance, that's why people began searching for a new technologies that is more feasible economically, these efforts led to what is known as Virtual Private Network (VPN).

Virtual private networks (VPNs) based on the Internet instead of the traditional leased lines offer organizations of all sizes the promise of a low-cost, secure electronic network. However, using the Internet to carry sensitive information can present serious privacy and security problems.

SSL VPN is an exciting new technology that allows remote access to applications and files from standard web browsers. Because they require no client-side software other than a web browser, SSL VPNs offers great convenience, and promise to provide a much lower total cost of ownership than IPSEC VPNs which is considered the most used type of VPNs.

The improvement in work efficiency covers all fields and aspects of life, from health care (as a consequence of improvement of networking) and commerce to education.

1.2 Problem Definition

To overcome the problems of security and privacy resulting from using a public network such as the internet VPNs use encryption and authentication techniques to provide a secure path for data in the internet, this may result in deterioration of the performance of the VPN.

Because of the importance of SSL VPN and how it is expected to change the world of networking a performance evaluation study for this technology is inevitable.

Estimating the effect of encryption and authentication techniques can give us an insight of the kind of suitable applications and thus choosing the right encryption and authentication technique for a certain scenario.

There are different encryption and authentication techniques which differ, mainly, in the key size and block size and thus in they differ in security and encryption cost. Choosing the encryption and authentication technique can affect the performance of the VPN greatly, for example a technique that results in low jitter and high transfer rate is suitable for real-time applications and so on.

1.3 Project Objective

The objective of this project is implementation and performance evaluation of an SSL VPN taking into consideration the effect of encryption and authentication technique on the performance of the VPN.

1.4 Thesis Layout

This thesis is organized as follows:

Chapter 2: Introduces the concepts of VPN and tunneling techniques and reviews the previous work in the topic by giving some examples of published researches.

Chapter 3: Describes project life cycle, hardware specifications and software programs used during phases of the project life cycle.

Chapter 4: Describes the implementation and performance evaluation of the SSL VPN by giving brief information about the configuration files, the testbeds, the performance evaluation criteria and experiments carried out to evaluate the implemented SSL VPN.

Chapter 5: Presents the results and analysis of the performance evaluation.

Appendix A: gives OpenVPN server configuration file

Appendix B: gives OpenVPN client configuration file

Appendix C: gives some necessary files to both NFS and ftp configuration

2 Literature Review

This chapter presents the necessary concepts and background information for a good understanding of the project by reviewing the current literature and discussing briefly the previous work in the area of VPN performance evaluation by giving some examples of published researches.

2.1 Virtual Private Network (VPN)

Virtual Private Network can be defined as "a way to simulate a private network over a public network, such as the Internet". It is called virtual because there are no real physical connections just routed packets over various machines on the internet. [1]

A virtual private network (VPN) is a very popular technology, and its popularity is increasing everyday especially among large organization that want to use the internet for both intra- and interorganization communication.

As mentioned earlier the aim of introducing VPNs is to minimize the cost required to maintain a wan, of course using the internet will do the job of minimizing the cost of networking but the reason that people didn't use the internet for this purpose was the security, performance and reliability. Security has always been a fundamental requirement of a network. Now internet is used to transfer very sensitive information in all fields, especially in the e-commerce field, which is a dominant field especially in the first world.

As for the network requirements, it is always a trade-off in choosing the most suitable type for a certain scenario and as mentioned earlier maintaining a WAN that is built on leased lines can be very expensive; taking all these facts into consideration VPNs can be a very tempting solution.

2.1.1 VPN types

VPNs can be deployed in different network infrastructures, in this section we will briefly explore some of the common situations in which a VPN may be deployed.

2.1.1.1 Site-to-Site VPNs

As cryptographic technology becomes more embedded in various network elements, growth in site-to-site VPN deployments has risen. A site-to-site VPN could be as simple as encrypting the link between two different nodes on a point-to-point connection.

Site-to-Site VPN connections are not only used to secure connectivity between two different organizations, but they are also used to secure links within an organization itself.

2.1.1.2 Remote Access VPNs

A user-to-LAN connection used by a company that has employees who need to connect to the private network from various remote locations. Typically, a corporation that wishes to set up a large remote-access VPN will outsource to an enterprise service provider (ESP). The ESP sets up a network access server (NAS) and provides the remote users with desktop client software for their computers. As home high-speed Internet access has increased throughout the world with the advent and deployment of cable modem technologies and DSL technologies, more companies are turning to RAVPN solutions to allow their workforce to establish secure connectivity to central corporate resources from remote locations. [2]

2.2 Tunneling

Tunneling is “the act of encapsulating the data from one protocol in the data of a protocol at the same or higher layer.The encapsulating protocol is said to be a tunnel for the lower-layer protocol” [3]. Providing a secure path for data through a public network is known as tunneling

2.2.1 Cryptography and Authentication

To be able to establish a secure path through the internet data must be protected, this protection is done through encryption and authentication.

Encryption is “the process of encoding information in such a way that only the person (or computer) with the key can decode it”. [3]

Most computer encryption systems belong in one of two categories:

- Symmetric-key encryption
- Public-key encryption

Symmetric-Key Encryption

Encryption algorithms that use the same key for encrypting and for decrypting information are called symmetric-key algorithms. The symmetric key is also called a secret key because it is kept as a shared secret between the sender and receiver of information.

A symmetric key between two parties is useful if it is used only once; it must be created for one session and destroyed when the session is over.

Public-Key Encryption

Encryption algorithms that use different keys for encrypting and decrypting information are most often called public-key algorithms but are sometimes also called asymmetric key algorithms. Public key encryption requires the use of both a private key (a key that is known only to its owner) and a public key (a key that is available to and known to other entities on the network). A user's public key, for example, can be published in the directory so that it is accessible to other people in the organization. The two keys are different but complementary in function.

Information that is encrypted with the public key can be decrypted only with the corresponding private key of the set.

One protocol, the Diffie-Hellman (DH) protocol, devised by Diffie and Hellman, provides a one-time session key for the two parties. The two parties use the session key without having to remember or store the key for the future.

Block Cipher

A symmetric encryption algorithm in which a block of plaintext bits (typically 64 or 128) is transformed as a whole into a cipher text block of the same length. Examples of block ciphers are Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Blowfish. .

Authentication

Authentication techniques are essential to VPNs, as they ensure that the communicating parties they are exchanging data with the correct user or host.

Digital Signature

Message authentication means that we must be sure that an imposter hasn't sent the message, message integrity means that data must arrive at the receiver as they were sent, and message nonrepudiation means that the receiver must be able to prove that a message has come from a specific user, these three services can be achieved by what is known as digital signature. [4]

A digital signature is "a type of asymmetric cryptography used to simulate the security properties of a handwritten signature on paper" [1]

Public Key Infrastructure

We will not be able to define the Public Key Infrastructure (PKI) unless we introduce the concept of digital certificates and Certificate Authority (CA).

When distributing public keys we must have some method to ensure that it is authentic, this is done using digital certificates.

The definition of a digital certificate leads us to a question of "who is going to issue these digital certificates?"

A certificate authority or certification authority (CA) is "an entity which issues digital certificates for use by other parties". [1]

A 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, which, depending on the level of assurance the binding has, may carried out by software at a CA, or under human supervision. 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 and other attributes are made unforgeable in public key certificates issued by the CA. [1]

Three major attributes of a successful security system are message authentication, message integrity and nonrepudiation.

Message Authentication Code (MAC)

A function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

Hash Function

A function that maps a message of any length into a fixed-length hash value, which serves as the authenticator. Examples of hash functions are Secure Hash Algorithm (SHA) and Message Digest 5 (MD5).

2.2.2 Tunneling Protocols

VPN tunneling is done using many protocols such as PPTP, IPSEC and SSL. A brief will be given brief information about IPSEC and SSL in the following sections.

2.2.2.1 IPSEC

IPSEC VPNs encrypt data at the Layer 3 IP packet layer, offering a comprehensively secure VPN solution through providing data authentication, anti-replay protection, data confidentiality, and data integrity protection. As such, IPsec is one of the most widespread VPN technologies in today's enterprise, service provider, and government networks.IPSEC consists of three major protocols:

- AH: A protocol that provides data origin authentication, data integrity, and replay protection.
- ESP: A protocol that provides the same services as AH but also offers data privacy through the use of encryption
- IKE : A protocol that provides the all-important key-management function. The alternative to IKE is manual keying, which IPSEC also supports

2.2.2.2 Secure Socket Layer (SSL)

The most used transport-layer tunneling protocol is the Secure Sockets Layer (SSL)the protocol used to, among other things, secure HTML (Hypertext Markup Language) transactions on the Web. SSL has many applications and can easily be used to build general-purpose transport-layer tunnels**.**

The first SSL specification originated in 1994 at Netscape, which was interested in a way to secure certain transactions made with its Netscape Navigator Web browser. The first version was not released outside Netscape. Later that same year, the specification for version 2 of SSL (SSL 2) was released, and an implementation appeared in Netscape Navigator 1.1 early in 1995. [3]

In 1996, the IETF began an effort to standardize the SSL protocol. For political reasons, the new protocol was named the Transport Layer Security (TLS) protocol. TLS is based mostly on version 3 of SSL but with enough "minor" changes to make it incompatible with SSL 3.

SSL makes use of three cryptographic functions. First, the two sides need a way of exchanging keying material with each other. Part of this key exchange can also provide authentication of the server.

Second, there must be a method of encrypting the application data and other secured messages in the protocol. SSL supports several ciphers, both stream and block, for this purpose.Finally, each record transmitted must be authenticated.

2.3 SSL VPN

In today's network architectures, Secure Socket Layer (SSL) VPNs represent one of the most popular choices for transport layer security.

SSL VPNs create secure tunnels by performing two functions:

- Requiring authentication from users before allowing access so that only authorized parties can establish tunnels
- Encrypting all data transmitted to and from the user by implementing the actual tunnel using SSL

SSL VPN technology allows users to remotely access important enterprise applications, systems, and files from standard web browsers.

SSL VPN technology promises to improve both employee productivity and convenience by freeing users from having to carry laptops when traveling, and allowing access from

any Internet-enabled computer. The technology also offers tremendous cost savings when compared to classic IPSEC VPNs since it does not require purchasing or maintaining remote client machines. Users also benefit, as they gain the convenience of being able to access corporate resources from any computer that they wish to use. [4]

2.4 Previous Work in the Topic

Because of the importance of SSL VPN and how much it is expected to change the world of networking, it has gained a lot of attention especially during the past 5 years, for example a conference was held in Paris, France in 2005 including a workshop named "SSL VPN securing remote access and extranets" discussing the methods for implementation and development of this technology [6], also a workshop held at Oxford university 2003 and many other workshops and conferences. [7]

Braun, et al (2000) did a performance evaluation study for an IPSEC VPN to estimate the effect of encryption and authentication techniques on the performance of the VPN by implementing a VPN between universities of Geneva and Bern. They found that packet loss isn't affected by encryption and authentication techniques but round trip time (RTT) increases and transfer rate using ftp decreases when applying encryption and authentication, and the heavier the amount of encryption the more they are affected.[8]
Khanvilkar and Khokhar, University of Illinois at Chicago (2004) did a performance evaluation study on Open-Source Linux-based VPNs (OSLVs) by measuring the following network measures:

- Jitter using Iperf
- Bandwidth using Iperf [9]

Their results can be summarized by the following figures:

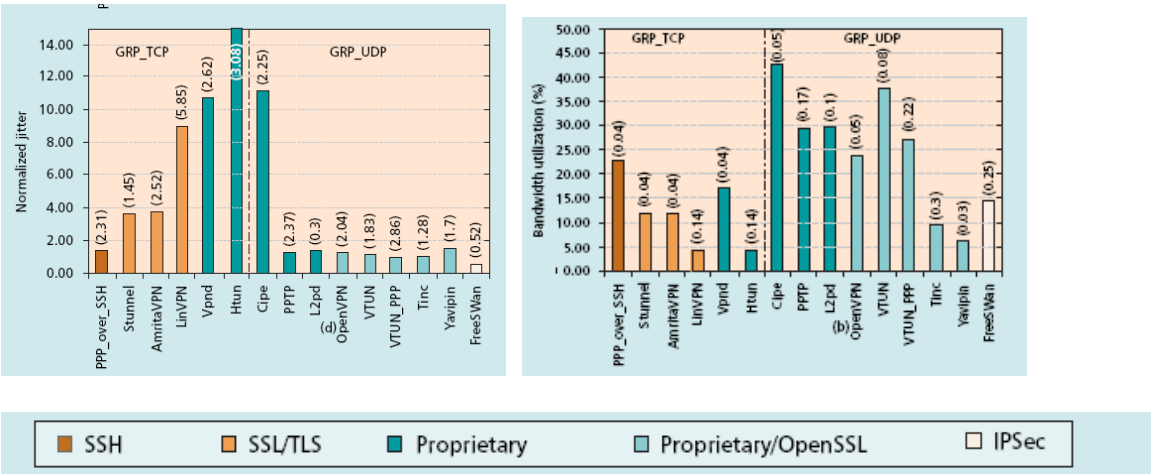


Figure 2.1 Normalized Jitter and Bandwidth

3 Materials and Methods

This chapter describes the hardware and software requirements needed for the implementation of the project.

3.1 Project life cycle

This section describes the several phases during the implementation of this project.

Preliminary Research Phase

In this phase, the available options for implementation and performance evaluation were studied and analyzed according to the design goals.

Implementation Phase

In this phase, SSL VPN was implemented, including the troubleshooting of the problems during the implementation.

Performance Evaluation Phase

In this phase the implemented SSL VPN was evaluated according to certain network measures, taking into consideration the effect of encryption and authentication techniques.

3.1.1 Implementation Phase

This section describes the specifications of hardware and software requirements needed during the implementation phase.

3.1.1.1 Hardware Specifications

The SSL VPN was implemented between two computers, using one as the server and the other as the client.

Server Specifications

Server hardware specifications are summarized in table 3.1.

Table 3.1 Server Hardware Specifications

Specification	Used Specification
Model	Intel(R) Pentium(R)4
CPU Speed	2.66GHz
RAM size	512 MB
Cache size	512 KB
Operating System	Red Hat Enterprise Linux 4
Kernel	2.6.9-5.EL

Client Specifications

Client hardware specifications are summarized in table 3.2 below.

Table 3.2 Client Hardware Specifications

Specification	Used Specification
Model	Intel(R) Pentium(R)4
CPU Speed	2.80GHz
RAM size	240 MB
Cache size	1024 KB
Operating System	Red Hat Enterprise Linux 4
Kernel	2.6.9-5.EL

3.1.1.2 Software Specification

SSL VPN was implemented using an open-source solution called OpenVPN.

3.1.2 Performance Evaluation Phase

During performance evaluation phase several software programs were used, they are:

- Ping program: used to measure packet loss and round trip time (RTT)
- FTP: used to measure transfer rate
- NFS: used to measure access time
- Tcpdump : used along with NFS to measure access time
- NTP: used to synchronize server and client to be able measure access time
- Ttcp: used to measure the throughput
- Netperf: used to measure the throughput
- Iperf : used to measure the jitter and bandwidth of the implemented SSL VPN
- SPSS: used to analyze the results of the performance evaluation statistically.

4 System Implementation

This chapter describes system implementation by describing configuration files, testbeds, performance evaluation criteria and experiments done to measure system performance.

4.1 Design Goals

To choose between the available options for implementation design goals were set, which include minimum cost, hardware-independence, stability, full-featured SSL VPN solution, high quality of security and user convenience.

4.1.1 Minimum Cost

One of the main goals of implementation of our SSL VPN is minimum cost. Since one of the main benefits of a VPN (Virtual Private Network) is minimizing cost and a system that is successful technically but economically not feasible is considered a failure, Cost should be considered as a priority.

4.1.2 Hardware-independence

With the development in all fields and the increase in number of vendors and customers sharing parts of the products mostly, the need for standardization has become inevitable that's why many organizations responsible for standardization were introduced such as ITU (International Telecommunication Union) and ISO(International Organization for Standardization).

In system design and Implementation, for the implemented system to be sound it should be hardware-independent (it shouldn't depend on the computer used or LAN card) to fulfill the requirement of standardization.

4.1.3 Stability

Since a performance evaluation study will be carried for the implemented SSL VPN, it should be stable, in other words it should have a stable performance under certain conditions.

4.1.4 Full-featured SSL VPN Solution

Some SSL VPN solutions have limited capabilities compared to other ones; they can tunnel only a limited package of applications, an example of these solutions SSL-explorer. This project targets to implement a full-featured SSL VPN.

4.1.5 High Quality of Security

Since security is a main concept in the technology of VPN, the implemented SSL VPN should have a high quality of security.

4.1.6 User Convenience

As it is said "Security's worst enemy is complexity" [10].When implementing an SSL VPN security should be considered as a priority and as a consequence complexity should be avoided.

4.2 System Implementation Options

To be able to choose the right option of implementation we should be aware of the advantages and disadvantages of these options, when implementing an SSL VPN there are two options: appliance-based SSL VPN and SSL VPN software products.

4.2.1 Appliance

Appliances is "a "black box" that function without requiring administrators to understand their internal workings" [11].Therefore they minimize a great deal of human errors and overhead costs during installation, configuration and maintaining an IT system. On the other hand, some isolation from internal technology definitely exits.

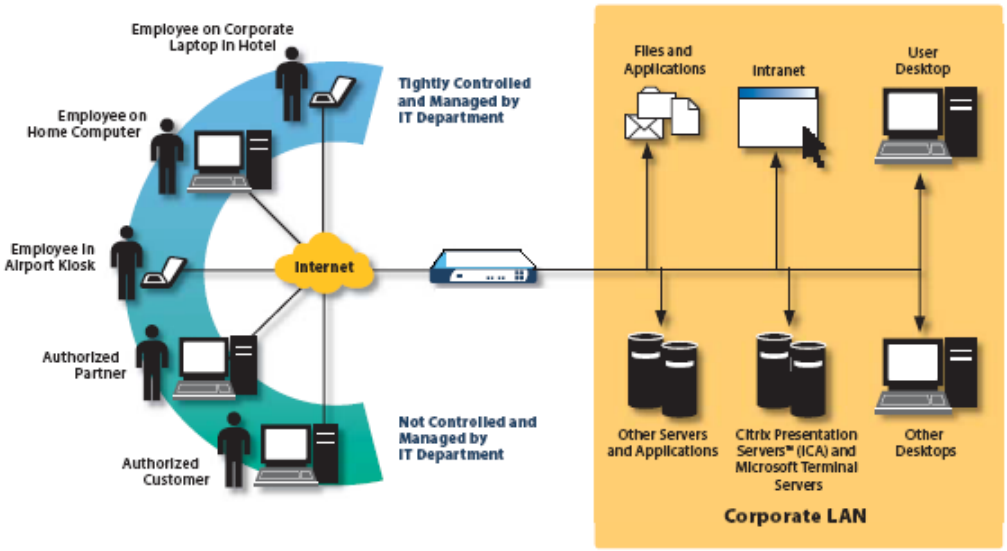


Figure 4.1 Appliance-based SSL VPN

4.2.2 SSL VPN software products

SSL VPN software products are sometimes preferred over appliance-based SSL VPN solutions especially in cases of organizations with data-center standards dictate that preferred brands of servers this is especially true in situations in which administrators are already skilled in hardening systems. [11]

Another advantage of SSL VPN software products is cost. Obviously using an SSL VPN software will save money because an appliance has both software and hardware components.

Since this project targets to do a performance evaluation for SSL VPN technology, an SSL VPN software product will be used to avoid the technology isolation caused by using an appliance-based SSL VPN. Another reason is project's budget.

4.3 OpenVPN

OpenVPN is a full-featured SSL VPN solution, which is available as an open-source. OpenVPN implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or 2-factor authentication, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN is not a web application proxy and does not operate through a web browser. [12]

4.4 OpenVPN Advantages

- OpenVPN's principal strengths include cross-platform portability across most of the known computing universe, excellent stability, scalability to hundreds or thousands of clients, relatively easy installation, and support for dynamic IP addresses and NAT.
- OpenVPN uses an industrial-strength security model designed to protect against both passive and active attacks. OpenVPN's security model is based on using SSL/TLS for session authentication and the IPSec ESP protocol for secure tunnel transport over UDP. OpenVPN supports the X509 PKI (public key infrastructure)

for session authentication, the TLS protocol for key exchange, the OpenSSL cipher-independent EVP interface for encrypting tunnel data, and the HMAC-SHA1 algorithm for authenticating tunnel data.

- OpenVPN is built for portability. At the time of this writing, OpenVPN runs on Linux, Solaris, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Windows 2000/XP. Because OpenVPN is written as a user-space daemon rather than a kernel module or a complex modification to the IP layer, porting efforts are dramatically simplified.
- OpenVPN has been built with a strongly modular design. All of the crypto is handled by the OpenSSL library, and all of the IP tunneling functionality is provided through the TUN/TAP virtual network driver.

4.5 OpenVPN Configuration file

One of the qualities of OpenVPN is configuration because it eliminates the complexity found in many other VPN solutions.

To implement an SSL VPN using OpenVPN, server and clients should be configured properly. Configuration should suit the requirement of the implemented scenario; therefore understanding the available options is the main key in configuring a successful SSL VPN using OpenVPN.

OpenVPN offers a great deal of options for both client and server including the kind of virtual network drive, transmission protocol (TCP and UDP), routed IP tunnel or bridged tunnel, using certificates, cipher algorithm used, message digest and many other options.

4.5.1 OpenVPN server configuration file

As it is said before, to configure OpenVPN you should be aware of the requirements of your network and choose the suitable options.

A brief explanation of the server configuration file used in the project will be given.

Refer to appendix A.

Service Port

Port 1194 was used for OpenVPN service. Other port could also be used.

Transmission Protocol

As for the transmission protocol UDP was used because there are problems when you tunnel TCP over TCP. TCP keeps track of packet sequence and packet loss and requests that missing packets be resent, which is a good thing when you only have one layer of TCP. It also has adaptive timers that dictate how long it will wait before it requests resends. This interval changes and basically increases exponentially as failures to receive packets continue. If you have TCP riding on top of TCP, you now have two flow control layers that are each providing timers and resend requests. If things line up poorly, for instances the "lower TCP layer" has a longer interval than your "higher layer" you can get a build up of requests from above that cause an internal meltdown in your flow control system. You end up slowing your TCP connection down to a crawl as redundant layers of flow control work against each other in an attempt to get packets resent. [12]

Virtual Drive

As for the virtual network drive tun virtual device was used because an IP routed tunnel was implemented.

A routed VPN creates a virtual subnet for the client and routes traffic over the VPN if addressed to a system on the remote side. A bridged VPN connects clients to an existing network subnet. Data is broadcast through the bridge like a virtual Ethernet hub.

Root Certificate

Authentication between server and clients is guaranteed through the use of certificates. As explained in chapter 2, the need for a certificate authority is inevitable to bind public keys and clients and the server.

In our server configuration file a root certificate was created and named ca.crt.

Client Key/Certificate Pair

In our scenario there is only one client. A pair of key/certificate was created for this client and named uofk.key and uofk.cert respectively.

Generation of Diffie-Hellman (DH) Parameters

1024 bit keys for Diffie-Hellman parameters were used during our test sessions

Virtual IP Addresses

Virtual network id used was: 10.8.0.0 subnet mask: 255.255.255.0

Server IP address 10.8.0.1

The Keepalive Directive

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 assumes that remote peer is down if no ping received during a 120 second time period.

Cryptographic Cipher

In this project three cryptographic ciphers were used which are: Blowfish, Triple-DES and AES.

Hash Algorithm

In this project three hash algorithms were used which are: Secure Hash algorithm (SHA), Secure Hash algorithm 1 (SHA 1) and Message Digest 5 (MD5).

Compression on the VPN Link

Compression in the VPN link was enabled to minimize the bandwidth required for the packets, which is fundamental requirement in most real-time scenarios.

Privilege Downgrade

When restarting a the computer privilege downgrade is forced to protect the system, this may cause a problem with some applications trying to access some resources that is unavailable because of privilege downgrade. OpenVPN configuration offers a directive to deal with this problem; it was used in the configuration.

Status File

OpenVPN offers a directive that outputs a status file showing current connections, rewritten every minute. Since a performance evaluation study was performed for the implemented SSL VPN this directive was used.

Status File Verbosity

File verbosity is controlled through a directive which specifies the level of verbosity, verb 5 was used because it can help in debugging connection problems.

4.5.2 Client configuration file

OpenVPN client configuration file is very similar to the server configuration, a brief explanation will be given for the client configuration file and client configuration file is included in Appendix B.

Client Declaration

First client declaration is needed and as a consequence it will need to pull certain configuration file directives from the server.

Virtual Drive

Virtual drive used is tun because an IP routed tunnel was implemented as explained in the server configuration file.

The Keepalive Directive

The keepalive directive was used.

Transmission Protocol

As in the server configuration file, transmission protocol used is UDP.

The Hostname/IP and Port of the Server

Server hostname or IP address must be specified in the client configuration file so that that the tunnel can be established through the internet.

Resolving the Host Name of the OpenVPN Server

When using the hostname of the OpenVPN server, we need to resolve it into the correspondent IP address, OpenVPN offers a directive to specify the amount of time the client will try to resolve the server hostname, infinite retrial time was chosen, because internet is unreliable.

Privilege Downgrade

Privilege downgrade upon restarting was used.

Root Certificate

Root certificate is common to all clients and should be indicated in every client configuration file, in the configuration it was named ca.crt, which stands for certificate authority.

Client Key/Certificate Pair

Every client needs a key/certificate pair for encryption and authentication purposes.

Cryptographic Cipher

In this project three cryptographic ciphers were used. They are triple-DES, AES and blowfish.

Hash Algorithm

In this project three hash algorithms were used. They are: Secure Hash algorithm (SHA), Secure Hash algorithm 1 (SHA 1) and Message Digest 5 (MD5).

Compression on the VPN Link

Compression was enabled on the VPN link.

Status File Verbosity

Level 5 was chosen of file verbosity.

4.6 Performance evaluation

Performance evaluation is very important to any system because it determines whether the system is sound or not, in addition to this it explains the kinds of applications suitable for the system. It is carried out according to many characteristics of the studied system and it differs according to the characteristics studied.

In this project a performance evaluation study for the implemented SSL VPN is done, which is very important since it is a promising technology expected to change the world of networking in many ways, and it has already had.

4.6.1 Performance Evaluation Criteria

To evaluate the implemented SSL VPN some tests were done measuring some characteristics of the network, which are

- round trip time (RTT)
- packet loss
- Jitter
- Throughput
- Bandwidth
- transfer rate
- Access time using NFS server.

Since a performance evaluation is done for the implemented SSLVPN, tests results were determined statistically.

The study was done taking into consideration the effect of encryption and authentication techniques on the performance of the SSL VPN, so different combinations of some of the cryptographic ciphers and hash algorithms were tried. The tests were done for each of these combinations; each test was done twenty times.

The cryptographic ciphers used in the VPN tunnel are: Triple-DES (Data Encryption Standard), blowfish, AES (Advanced Encryption Standard).

Hash Algorithms used are: SHA (Secure Hash function), SHA 1, MD5 (Message Digest 5).

4.7 Test-beds

Tests were done in university of Khartoum, Faculty of Engineering and Architecture labs. Some tests were done in Electronics lab and the rest in Networks lab. A brief description of the two test-beds will be given in the following sections.

4.7.1 First Testbed

First testbed is communications lab, faculty of engineering, university of Khartoum. Figure 4.3 below describes the testbed

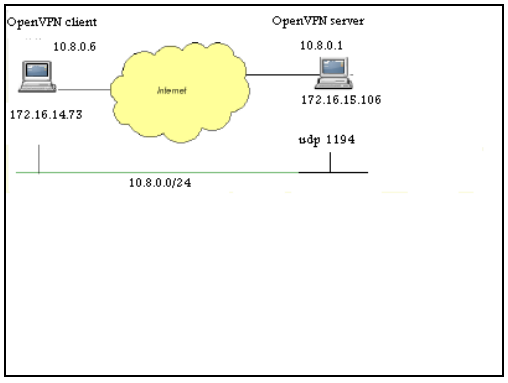


Figure 4.2 First Testbed

Table 4.1 below shows some specifications of the first testbed

Table 4.1 First Testbed Specifications

Specification	Used Specification
Connecting Device	Intel Express 10/100 Hub
Internet Bandwidth	2 Mbps (Shared among University of Kh. Center campuses)
Cables	Cupper

4.7.2 Second Testbed

In the second testbed same computers were used with the same LAN cards so the same IP address were used. Also same virtual IP addresses were used. The second testbed differs from the first in some specifications summarized in table 4.2.

Table 4.2 Second Testbed Specifications

Specification	Used Specification
Connecting Device	Huawei Quidway S2000-EI Intelligent Access L2 Switches
Internet Bandwidth	5 Mbps (Shared among University of Kh. campuses)
Cables	Fiber Optic

4.8 Tests

To do a performance evaluation for the implemented SSL VPN some network measures were calculated, each of which were done twenty times for each combination of cryptographic cipher and hash algorithm, as mentioned earlier. The combinations are: BF-SHA, BF-SHA1, BF-MD5, DES-SHA, DES-SHA1, DES-MD5, AES-SHA, AES-SHA1 and AES-MD5. In this section a brief description of the tests carried out will be given.

4.8.1 Ping

While there are several useful programs for analyzing connectivity, unquestionably ping is the most commonly used program.

One network device sends a request for a reply to another device and records the time the request was sent. The device receiving the request sends a packet back.

When the reply is received, the round-trip time for packet propagation can be calculated. The receipt of a reply indicates a working connection. This elapsed time provides an indication of the length of the path. Consistency among repeated queries gives an indication of the quality of the connection. [13]

We used ping to calculate the Round Trip Time (RTT) and packet loss in the implemented network.

Figure 4.2 shows ping working

```
[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=0.806 ms
64 bytes from 10.8.0.1: icmp_seq=1  ttl=64 time=0.711 ms
64 bytes from 10.8.0.1: icmp_seq=2  ttl=64 time=0.699 ms
64 bytes from 10.8.0.1: icmp_seq=3  ttl=64 time=0.824 ms
64 bytes from 10.8.0.1: icmp_seq=4  ttl=64 time=0.701 ms
64 bytes from 10.8.0.1: icmp_seq=5  ttl=64 time=0.685 ms
64 bytes from 10.8.0.1: icmp_seq=6  ttl=64 time=0.708 ms
64 bytes from 10.8.0.1: icmp_seq=7  ttl=64 time=0.703 ms
64 bytes from 10.8.0.1: icmp_seq=8  ttl=64 time=0.720 ms
64 bytes from 10.8.0.1: icmp_seq=9  ttl=64 time=0.696 ms
64 bytes from 10.8.0.1: icmp_seq=10 ttl=64 time=0.726 ms
64 bytes from 10.8.0.1: icmp_seq=11 ttl=64 time=0.697 ms
64 bytes from 10.8.0.1: icmp_seq=12 ttl=64 time=0.697 ms
```

Figure 4.3 Ping Used for Measuring RTT and Packet Loss

4.8.2 Throughput

Throughput is “a measure of the amount of data that can be sent over a link in a given amount of time”. [13]

Throughput estimates, typically obtained through measurements based on the bulk transfer of data, are usually expressed in bits per second or packets per second.

To measure the throughput of the implemented SSL VPN two software programs were used. They are called TTCP and NETPERF to compare the results obtained by each of them.

4.8.2.1 TTCP for throughput measurement

TTCP must be installed in both client and server, and then a communication channel between them must be opened by giving some commands, on both systems and referring to the server on the client side.

Figure 4.3 below show how TTCP works.

```
[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 3.672 real seconds = 4461.665 KB/sec +++
ttcp-t: 2048 I/O calls, msec/call = 1.836, calls/sec = 557.708
ttcp-t: 0.000user 0.145sys 0:03real 3% 0i+0d 0maxrss 0+4pf 449+31csw
[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 3.726 real seconds = 4397.432 KB/sec +++
ttcp-t: 2048 I/O calls, msec/call = 1.863, calls/sec = 549.679
ttcp-t: 0.003user 0.157sys 0:03real 4% 0i+0d 0maxrss 0+4pf 447+56csw
```

Figure 4.4 Throughput Test Using TTCP

Figure 4.3 shows a example of a testing network throughput using TTCP. On the client side, a command was given referring to the server holding the virtual ip address 10.8.0.1, in the figure the test was done twice giving results of 4461.665 KB/sec and 4397.432 KB/sec.

The above test was done for different combinations of cryptographic ciphers and hash algorithms in the SSL VPN tunnel, each combination was tested twenty times.

4.8.2.2 NETPERF for throughput measurement

NETPERF is a tool used for measuring network performance, originally designed at the Information Networks Division of Hewlett-Packard.

To measure throughput using NETPERF it must be installed in both systems (server and client) and then a communication channel between them must be opened by giving some

commands, on both systems and referring to the server on the client side, the same procedure used with TTCP.

Figure 4.4 shows NETPERF working in the 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    10.01      47.00
[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      51.94
```

Figure 4.5 NETPERF Used for Throughput Measurement

In figure 4.4, NETPERF is used to measure the throughput in the client side, referring to the server holding the virtual IP address 10.8.0.1 and communicating through port 1194, which is OpenVPN port. We notice that the test was done twice giving results of 47.00 10^6bits/sec and 51.94 10^6bits/sec for the throughput.

4.8.3 Bandwidth

Bandwidth measurements will give you an idea of the hardware capabilities of your network, such as the maximum capacity of your network.

The bandwidth was measured using a software program called Iperf. Iperf was designed at the National Laboratory for Applied Network Research (NLANR).
As in the case with Ttcp and Netperf, must be installed in both systems (server and client) and then a communication channel between them must be opened by giving some commands, on both systems and referring to the server on the client side and the port of operation should be specified, in our case it is 1194 (OpenVPN port).

In the figure below, an example showing Iperf running on the client side of our SSL VPN.

```
[root@localhost ~]# iperf -c10.8.0.1 -p1194
------------------------------------------------------------
Client connecting to 10.8.0.1, UDP port 1194
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 10.8.0.6 port 1194 connected with 10.8.0.1 port 1194
[  3]  0.0-10.0 sec  56.9 MBytes  47.7 Mbits/sec
[root@localhost ~]# iperf -c10.8.0.1 -p1194
------------------------------------------------------------
Client connecting to 10.8.0.1, UDP port 1194
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 10.8.0.6 port 1194 connected with 10.8.0.1 port 1194
[  3]  0.0-10.0 sec  51.2 MBytes  42.8 Mbits/sec
```

Figure 4.6 Iperf Running

In the above figure, we notice that Iperf needs two ports for the communication channel to be established between the two systems to be established, a local port (could be any unused port) and a remote port (1194 OpenVPN port in our case).We notice the virtual IP address of the client and server : 10.8.0.6 and 10.8.0.1 respectively.
Bandwidth was measured for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each.

4.8.4 Jitter

Network jitter term is associated with the loss or desequencing of data packets in a real-time data stream. Lost audio packets are not recovered and would therefore produce some form of audio loss. Since audio operates with smaller packets at a lower bandwidth, in general, it is usually less likely to encounter network jitter, but an audio stream is not immune from the effects of jitter. [14]

Measuring the jitter of a network can give us an insight of the kind of applications that can be done using that network, that's why jitter was chosen as one of the parameters for evaluating the implemented SSL VPN in this project.

Jitter was measured using Iperf, the same program used for measuring the bandwidth.

The jitter was measured for the implemented SSL VPN for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each.

The figure below shows Iperf running for measuring the jitter, on the client side of the implemented SSL VPN.

```
[root@localhost ~]#  iperf -s -u -i 1
------------------------------------------------------------
Server listening on UDP port 1194
Receiving 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 10.8.0.1 port 1194 connected with 10.8.0.6 port 1194
[  3]  0.0- 1.0 sec  1.19 MBytes  10.0 Mbits/sec  0.019 ms    0/  850
(0%)
[  3]  1.0- 2.0 sec  1.19 MBytes  10.0 Mbits/sec  0.014 ms    0/  851
(0%)
[  3]  2.0- 3.0 sec  1.19 MBytes  10.0 Mbits/sec  0.024 ms    0/  850
(0%)
[  3]  3.0- 4.0 sec  1.19 MBytes  10.0 Mbits/sec  0.019 ms    0/  850
(0%)
[  3]  4.0- 5.0 sec  1.19 MBytes  9.98 Mbits/sec  0.015 ms    0/  849
(0%)
[  3]  5.0- 6.0 sec  1.19 MBytes  10.0 Mbits/sec  0.268 ms    0/  852
(0%)
[  3]  6.0- 7.0 sec  1.19 MBytes  10.0 Mbits/sec  0.167 ms    0/  850
(0%)
[  3]  7.0- 8.0 sec  1.19 MBytes  10.0 Mbits/sec  0.011 ms    0/  851
(0%)
[  3]  8.0- 9.0 sec  1.19 MBytes  10.0 Mbits/sec  0.039 ms    0/  850
(0%)
[  3]  9.0-10.0 sec  1.19 MBytes  10.0 Mbits/sec  0.018 ms    0/  850
(0%)
```

Figure 4.7 Iperf Measuring Jitter

4.8.5 Ftp test

File Transfer Protocol (FTP) is a fundamental internet service, one that almost every internet user has used it at one time or another. It is used to transfer files between systems in a network. [15]

To measure the transfer rate of our network configured Ftp was configured in our SSL VPN, and transfer a file of size 33 KB then ftp gives the transfer rate.

The transfer rate was measured using Ftp server for the implemented SSL VPN for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each. Figure 4.7 below shows ftp used for transferring a file called openvpn of size 33 KB and type .PDF.

```
[root@localhost ~]# ftp 10.8.0.1
Connected to 10.8.0.1.
220 (vsFTPd 2.0.1)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (10.8.0.1:root): sslvpn
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get openvpn.pdf
local: openvpn.pdf remote: openvpn.pdf
227 Entering Passive Mode (10,8,0,1,197,114)
150 Opening BINARY mode data connection for openvpn.pdf (342434 bytes).
226 File send OK.
342434 bytes received in 0.33 seconds (1e+03 Kbytes/s)
```

Figure 4.8 Ftp Used for Transferring a File

4.8.6 Network File System (NFS) for measuring access time

NFS is the most common method used for sharing files across Linux and Unix networks. It is a distributed file system that enables local access to remote disks and file systems. NFS enables access to remote files and directories on other systems in the network that means you can edit and change these accessed files if you are authorized, which differs from Ftp which enables you to transfer the file, i.e. copy it to your system. [15]

NFS was configured in our implemented SSL VPN to measure access time, i.e. how much time we need to access a file.

To do this Tcpdump was used which is a packet sniffer that enables you to examine the traffic going on your network, or with a specific client.

Tcpdump was installed on both the client and server, and examined the access time for a specific packet of a file, and the file itself.

To do this we must synchronize both our systems. To do this a time server solution was used named Network Time Protocol (NTP).

4.8.6.1 Synchronization using Network Time Protocol (NTP)

NTP is an open standard that defines how Internet time servers work and how clients can communicate with these servers to maintain accurate time. [15]

NTP was configured on our implemented SSL VPN to synchronize our systems.

The test was done for each of the combinations of the cryptographic ciphers and hash algorithms.

Figure 4.8 below, shows Tcpdump being running, while NFS is being used to access a file.

```
[root@localhost ~]# tcpdump -i tun0 host 10.8.0.1
tcpdump: WARNING: arptype 65534 not supported by libpcap - falling back
to cooke d socket
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on tun0, link-type LINUX_SLL (Linux cooked), capture size 96
bytes
14:53:59.804969 IP 10.8.0.6.rndc > 10.8.0.1.sunrpc: S
2254617937:2254617937(0) w in 5840 <mss 1460,sackOK,timestamp 4876059
0,nop,wscale 2>
14:53:59.805476 IP 10.8.0.1.sunrpc > 10.8.0.6.rndc: S
2246724534:2246724534(0) a ck 2254617938 win 5792 <mss
1368,sackOK,timestamp 4898388 4876059,nop,wscale 2>
14:53:59.805497 IP 10.8.0.6.rndc > 10.8.0.1.sunrpc: . ack 1 win 1460
<nop,nop,ti mestamp 4876060 4898388>
14:53:59.805555 IP 10.8.0.6.rndc > 10.8.0.1.sunrpc: P 1:61(60) ack 1
win 1460 <n op,nop,timestamp 4876060 4898388>
```

Figure 4.9 Tcpdump running to capture traffic with server 10.8.0.1

In the figure, we notice that the virtual network drive tun0 has been specified in addition to the virtual IP address of the server 10.8.0.1, to make sure that traffic captured would be that of the SSL VPN.

5 Performance Evaluation

In this chapter the results obtained of the performance of the implemented SSL VPN for the different combinations of cryptographic ciphers and hash algorithms will be given and analyzed.

The following abbreviations were used to denote the different cryptographic ciphers and hash algorithms:

BF: Blowfish, DES: triple-Data Encryption Standard, AES: Advanced Encryption Standard, SHA: Secure Hash Algorithm, SHA1: Secure Hash Algorithm 1, MD5: Message Digest 5.

5.1 Packet loss

Packet loss is one of the key measures to evaluate a network. To calculate the packet loss in the implemented SSL VPN ping program was used. Fifty packets were sent each time and the lost packets were calculated. The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each; the results were analyzed statistically using the SPSS statistical analysis tool.

Figure 5.1 below shows the results of the test for each of the combinations

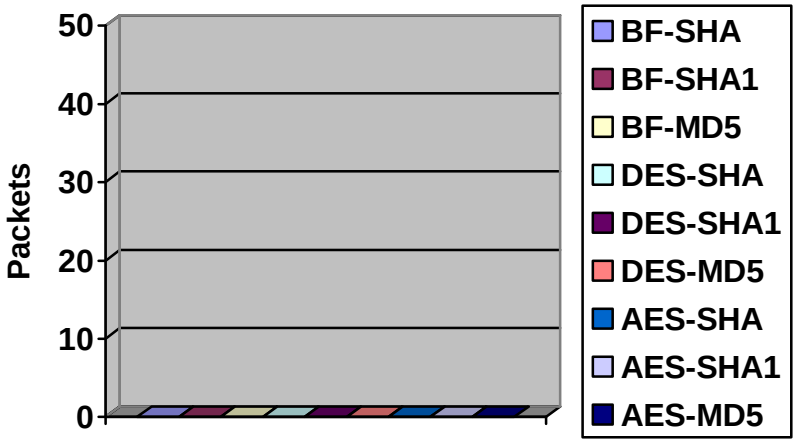


Figure 5.1 Packet Loss Results

Figure 5.1 shows the results of packet loss measurements; we notice that there are no packet losses for each of the combinations.

This result means that packet loss is independent of the combination applied on the VPN tunnel.

5.2 Round Trip Time (RTT)

To calculate the Round Trip Time (RTT) in the implemented SSL VPN, ping program was used. The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each.

In the following sections the results for the different combinations will be given then a comparison between them will also be given.

5.2.1 RTT Histograms for Different Combinations

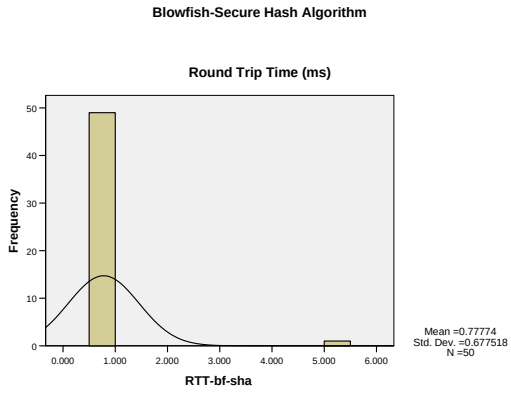


Fig. 5.2 RTT Histogram for BF-SHA

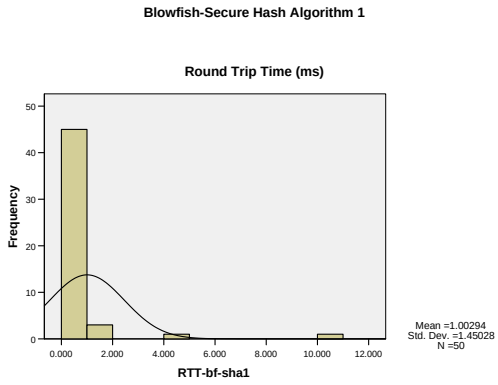


Figure 5.3 RTT Histogram for BF-SHA1

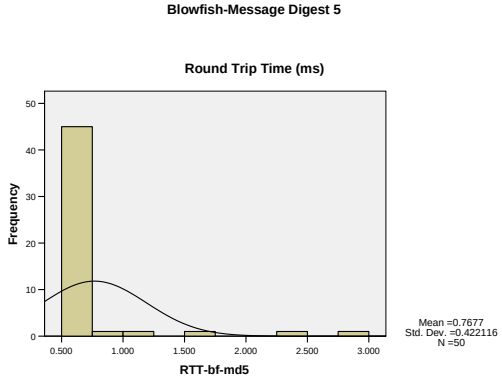


Figure 5.4 RTT Histogram for BF-MD5

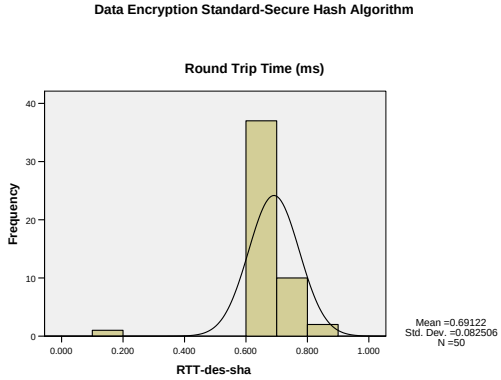
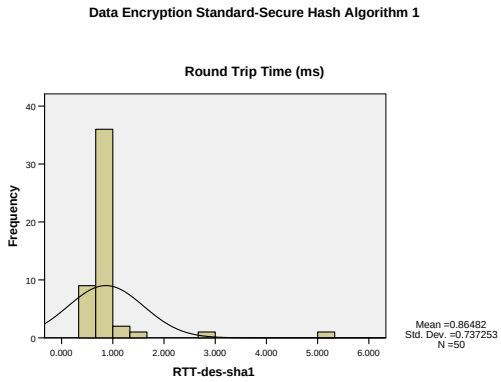


Figure 5.5 RTT Histogram for DES-SHA



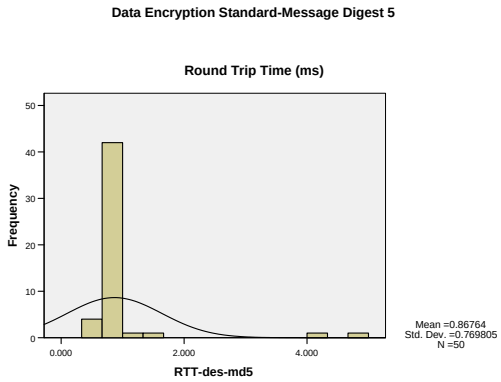
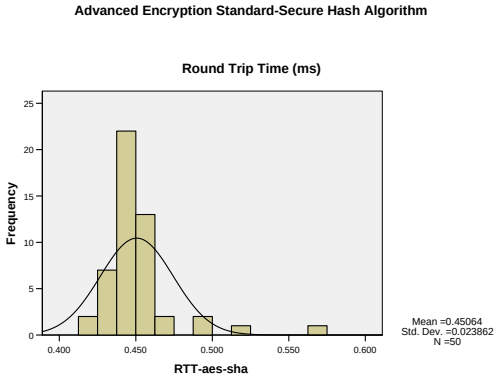


Figure 5.6 RTT Histogram for DES-SHA1 Figure 5.7 RTT Histogram for DES-MD5



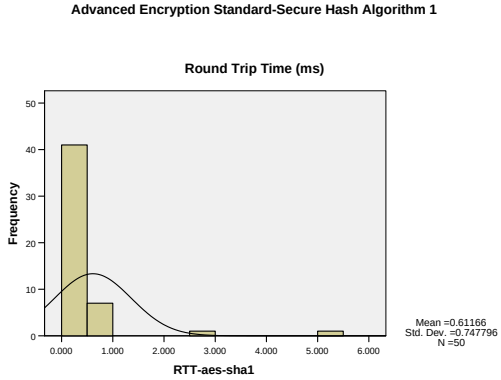


Figure 5.8 RTT Histogram for AES-SHA Figure 5.9 RTT Histogram for AES-SHA1

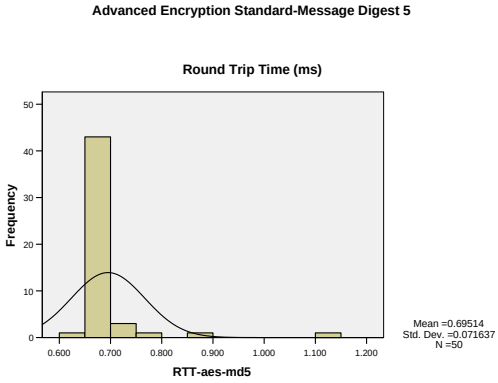


Figure 5.10 RTT Histogram for AES-MD5

5.2.2 Statistical Comparison of RTT Results

In this section the results of the mean RTT values for the different combinations will be compared statistically and analyzed.

Table 5.1 shows Ping RTT (ms) results of different combinations of cryptographic ciphers & hash algorithms being compared statistically.

Table 5.1 Statistical Comparison of RTT Results

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

In table 5.1, for each of the combinations the mean, maximum, minimum, standard deviation and variance were calculated. Standard deviation value indicates the ability of using the mean value as a reference for comparison, because for a large value of the standard deviation it would be inconvenient to use the mean as a reference value.

Figure 5.11 shows the mean RTT values of the different combinations of cryptographic ciphers and hash algorithms.

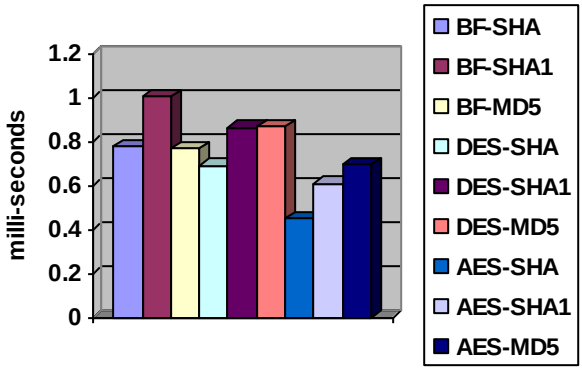


Figure 5.11 Mean RTT Results for different combinations

All of the combinations tests were done in the first testbed except AES-SHA, AES-SHA1 combinations.

We notice that the BF-SHA1 combination has the largest mean RTT value which is due to the RTT odd value (10 ms), referring to figure 5.2, the histogram shows that most values lie on the range of (0-1) ms, also from the statistical analysis results we notice that the BF-SHA1 has a standard deviation of 1.4 which is considered a large value in this case, which means we cannot take the RTT mean value in this case into consideration.

On the other hand, we notice that the AES-SHA has the least mean RTT value followed by the combination AES-SHA1 which may be explained by the difference in the Bandwidth of the internet used, which the VPN depends on.

Referring to figure 5.8 the histogram shows that most of the values lies in the range of (0-1) ms.The difference in mean values between the last two combinations may be explained by the odd value of 5.3 ms, as shown in figure 5.8.

From the above, changing the combination of cryptographic ciphers has negligible effect on the RTT value, but the testbed specifications such as internet Bandwidth and connecting device has a considerable effect on the RTT value.

5.3 Throughput Measured Using TTCP

Throughput was calculated using two different programs one of them is Ttcp.The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each. In the following sections the results for the different combinations will be given and analyzed.

5.3.1 TTCP Throughput Histograms for Different Combinations

In this section the Throughput histograms for different combinations will be given.

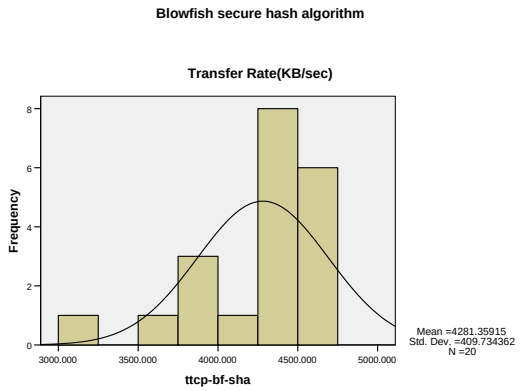


Figure 5.12 BF-SHA Ttcp Throughput

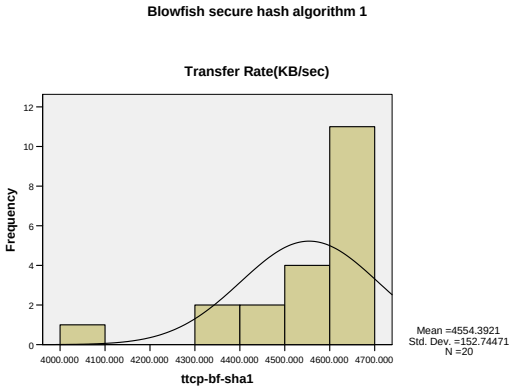


Figure 5.13 BF-SHA1 Ttcp Throughput

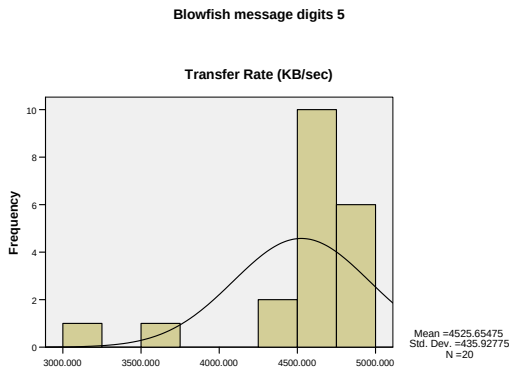


Figure 5.14 BF-MD5 Ttcp Throughput

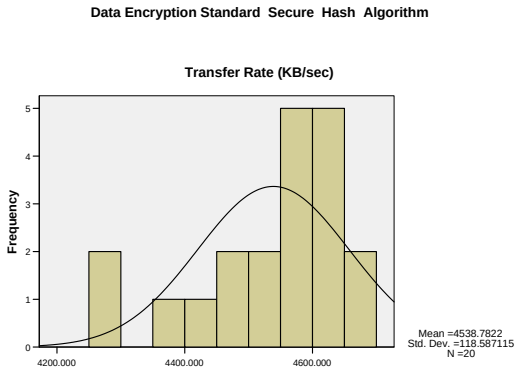


Figure 5.15 DES-SHA Ttcp Throughput

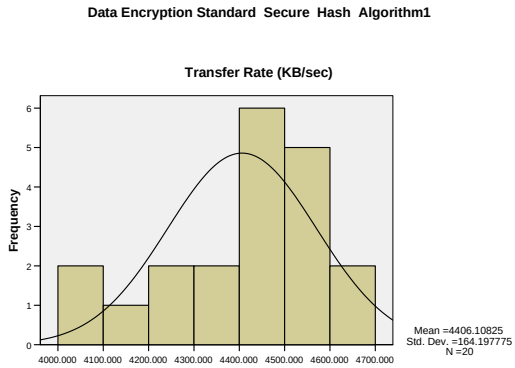


Figure 5.16 DES-SHA1 Ttcp Throughput

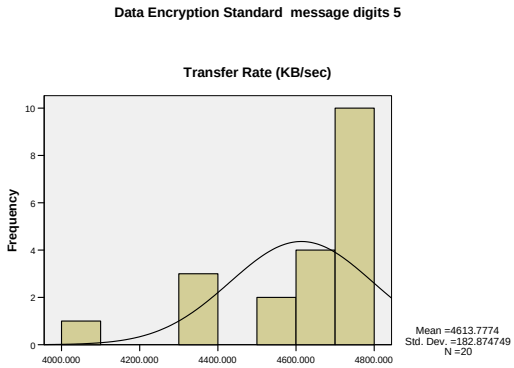


Figure 5.17 DES-MD5 Ttcp Throughput

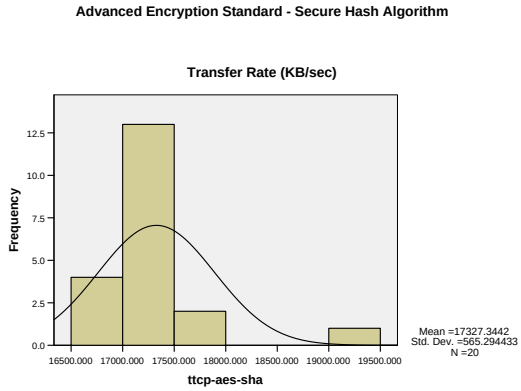


Figure 5.18 AES-SHA Ttcp Throughput

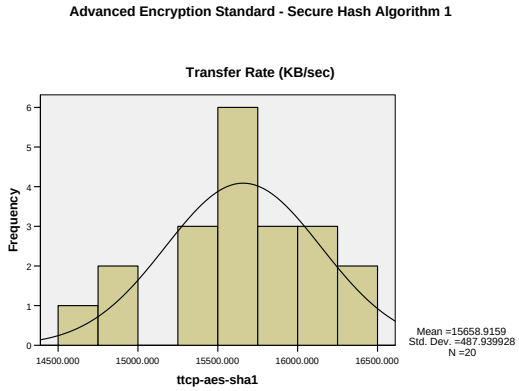


Figure 5.19 AES-SHA1 TtcpThroughput

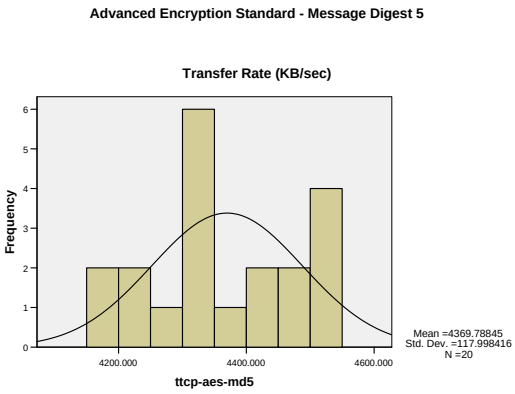


Figure 5.20 AES-MD5 Ttcp Throughput

5.3.2 Statistical Analysis of Ttcp Throughput Results

In this section the throughput results measured using Ttcp for the different combinations will be compared statistically and analyzed.

Table 5.2 below shows statistical analysis of the results of throughput measured using Ttcp.

Table 5.2 Statistical Analysis of Ttcp Throughput Results

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

In table 5.2, for each of the combinations the mean, maximum, minimum, standard deviation and variance were calculated. Standard deviation value indicates the ability of using the mean value as a reference for comparison, because for a large value of the standard deviation it would be inconvenient to use the mean as a reference value.

As said before, the tests for the combinations AES-SHA, AES-SHA1 were carried out in the second testbed, that's why they can't be included in this comparison but their results can be very useful in estimating the effect of testbed specifications in the performance of the VPN.

As indicated by the standard deviation column the combinations BF-SHA, BF-MD5 have relatively large standard deviation values which can be explained by the odd values of 3123.56 and 3015.14 as indicated by the histograms in figures 5.11 and 5.13 respectively.

Figure 5.21 below shows the mean values of throughput measured by Ttcp for different combinations.

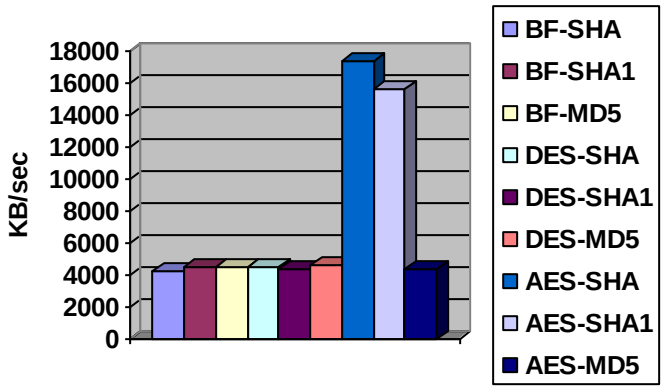


Figure 5.21 Mean Values of Ttcp Throughput for Different Combinations

Figure 5.21 shows that the AES-SHA1 and AES-SHA have the highest throughput which is about three times the throughput of the other combinations. This may be explained by the testbed specifications, because they were carried out in the second testbed. In which the bandwidth of the internet is 5 Mbps shared among University of Khartoum campuses using fiber optic cables which is higher than the 1.5 Mbps bandwidth shared among University of Khartoum center campus using cupper cables (which is slower than fiber optic cables) used in the first testbed. In addition to this the second testbed uses Huawei S2000 switch for connecting devices in the lab which is faster than the Intel Hub used in the first testbed because hubs are half-duplex devices whereas switches are full-duplex devices. [16]

Considering the rest of the combinations we notice that their mean throughput values lie in the range (4281.35-4613.77) KB/s. Referring to figure 5.12, the histogram shows that the mean value decrement is due to the odd value of 3123.56 therefore in this case the mean isn't considered a good reference for comparison.

To sum up, changing the combination of cryptographic cipher and hash algorithm has a negligible effect on the throughput measured by Ttcp which may be explained by the fact that throughput is concerned with the data transmitted excluding additional headers during different layers of OSI model and encryption and authentication techniques which is expected to have negligible effect in this case but it should have a considerable effect on transfer rate which considers the whole data packets including

headers of different layers of OSI model as will be seen, but the testbed specifications such as the bandwidth of the internet used and the device used can greatly affect the throughput.

5.4 Throughput Measured Using Netperf

Netperf program was also used to measure the throughput of the implemented SSL VPN to compare the results of throughput calculated by Ttcp and Netperf.The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each. In the following sections the results for the different combinations will be given and analyzed.

5.4.1 Netperf Throughput Histograms

In this section the Throughput histograms for different combinations will be given., we will use NP to denote Netperf

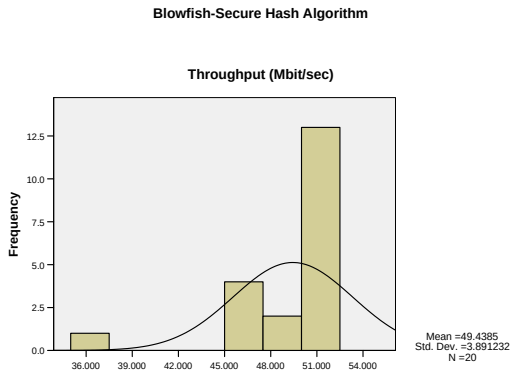


Figure 5.22 BF-SHA NP Throughput

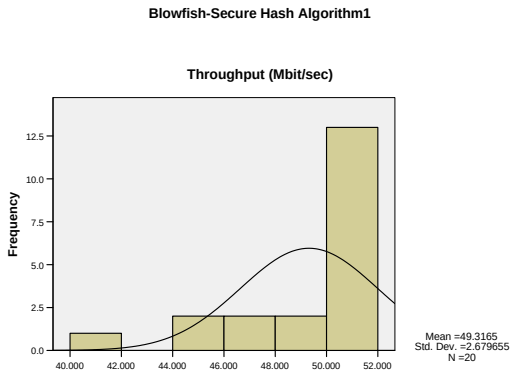


Figure 5.23 BF-SHA1 NP Throughput

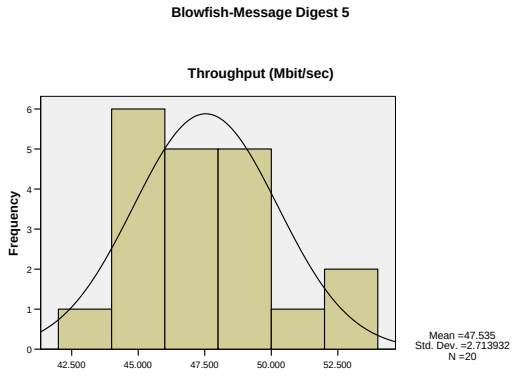


Figure 5.24 BF-MD5 NP Throughput

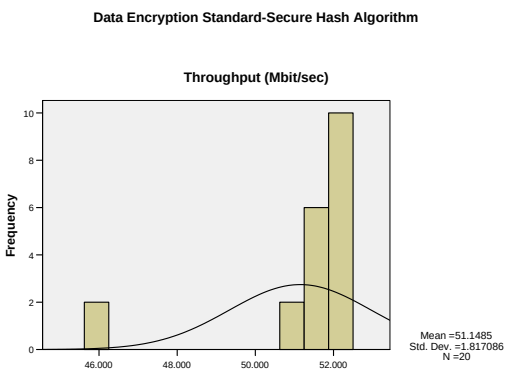
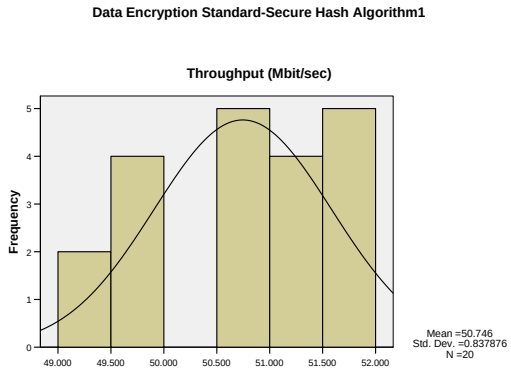


Figure 5.25 DES-SHA NP Throughput



5.26 DES-SHA1 NP Throughput

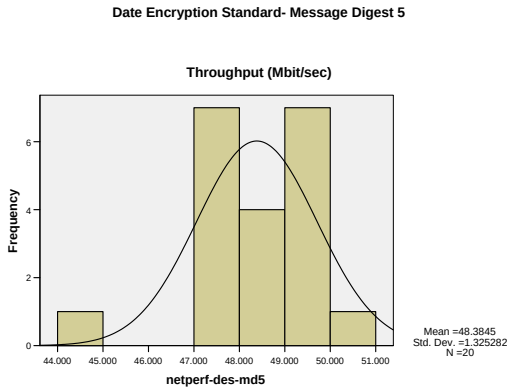
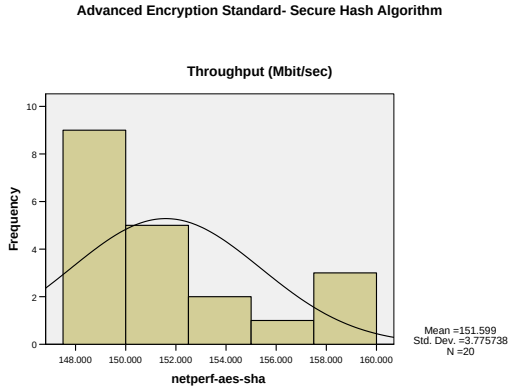
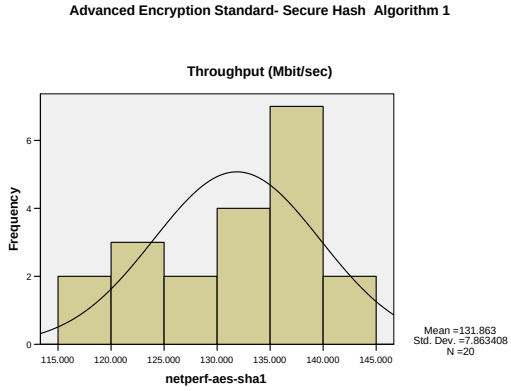


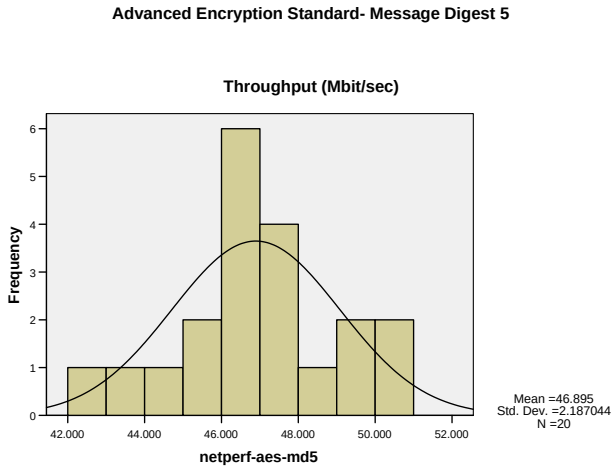
Figure 5.27 DES-MD5 NP Throughput



5.28 AES-SHA NP Throughput



5.29 NP Throughput AES-SHA1



5.30 Netperf Throughput Histogram for AES-MD5

5.4.2 Statistical Analysis of Netperf Throughput Results

In this section the throughput results measured using Ttcp for the different combinations will be compared statistically and analyzed.

Table 5.3 below shows statistical analysis of the results of throughput measured by Netperf for different combinations.

Table 5.3 Statistical Analysis of Netperf Throughput Results

Algorithim	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

In table 5.3, for each of the combinations the mean, maximum, minimum, standard deviation and variance were calculated.

Figure 5.31 below shows Throughput mean values for different combinations

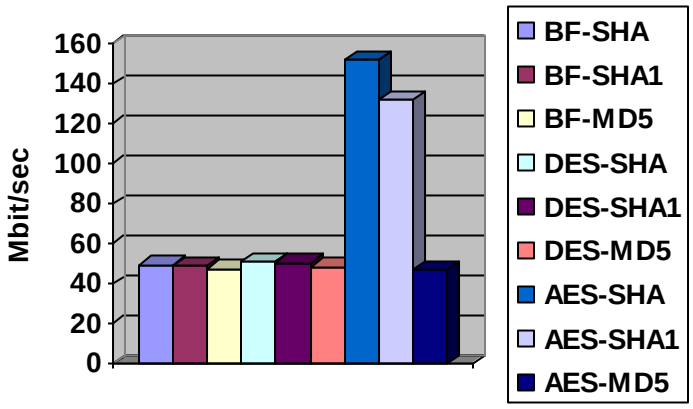


Figure 5.31 Comparison of NP Throughput Mean Values

As said before, the tests for the combinations AES-SHA, AES-SHA1 were carried out in the second testbed, that's why they can't be included in this comparison but their results can be very useful in estimating the effect of testbed specifications in the performance of the VPN.

Again we notice that the AES-SHA1 and AES-MD5 have the highest throughput which is about three times the throughput of the other combinations. This may be explained by the testbed specifications, because they were carried out in the second testbed. In which the bandwidth of the internet is 5 Mbps shared among University of Khartoum campuses using fiber optic cables which is higher than the 1.5 Mbps bandwidth shared among University of Khartoum center campus using cupper cables (which is slower than fiber optic cables) used in the first testbed. In addition to this the second testbed uses Huawei S2000 layer 2 device for connecting devices in the lab which is faster than the Intel express 10/100 stackable Hub used in the first testbed.

As for the rest of the combinations which were carried out in the first testbed we notice that DES-SHA and DES-SHA1 has the largest mean throughput values which may be explained by the size of the key used in triple DES cryptographic cipher (122 bits) which is less than key sizes for both blowfish cryptographic cipher (128 bits) and AES cipher (256 bits), which means that DES needs less computational effort than both blowfish and AES ciphers [17]. We can also notice that the combination AES-MD5 has the least throughput mean value, which may be explained by the computational effort needed for this algorithm. Generally, changing the combination doesn't have a considerable effect in changing the mean throughput measured by Netperf.

5.5 Bandwidth Measured Using Iperf

Iperf program was used to measure the Bandwidth of the implemented SSL VPN to compare the bandwidth results with the throughput calculated by Ttcp and Netperf. The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each. In the following sections the results for the different combinations will be given and analyzed.

5.5.1 Bandwidth Histograms for Different Combinations

In this section the bandwidth histograms for different combinations will be given.

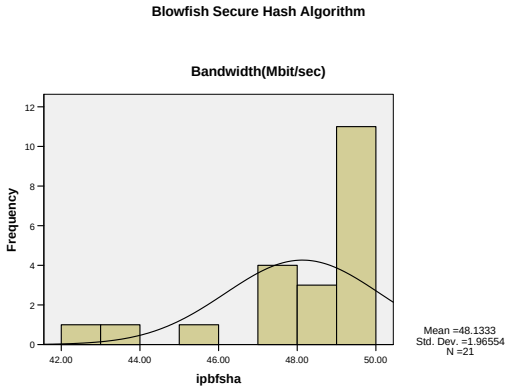


Figure 5.32 Bandwidth for BF-SHA

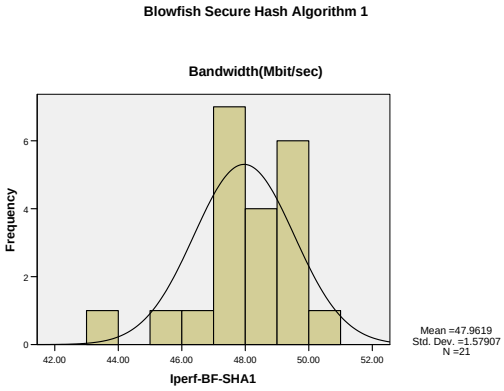


Figure 5.33 Bandwidth for BF-SHA1

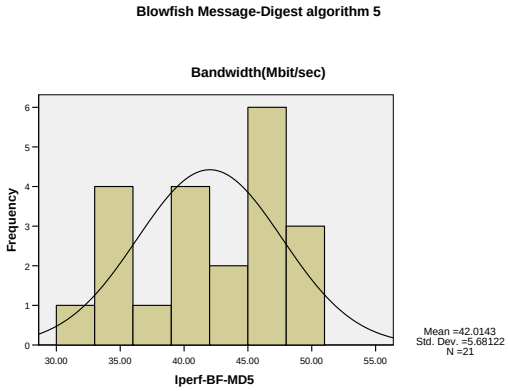


Figure 5.34 Bandwidth for BF-MD5

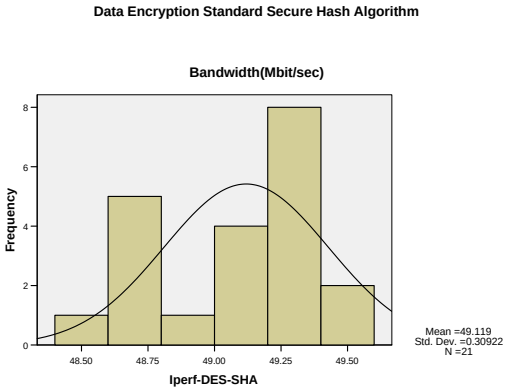


Figure 5.35 Bandwidth for DES-SHA

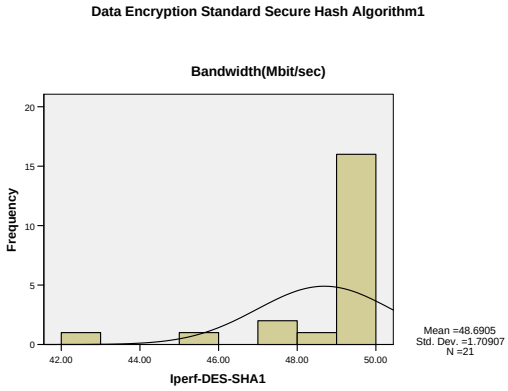


Figure 5.36 Bandwidth for DES-SHA1

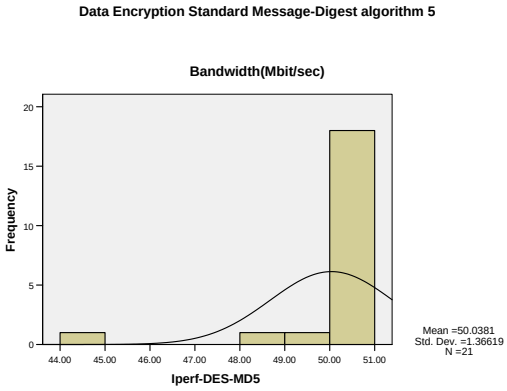
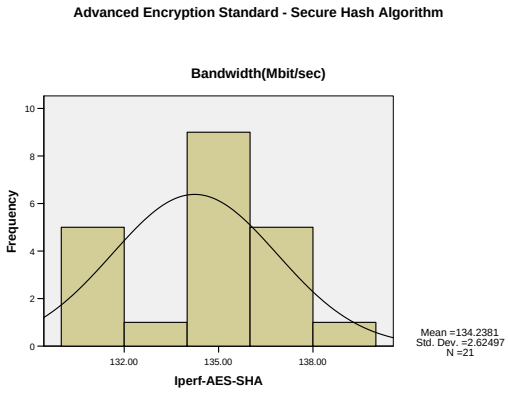


Figure 5.37 Bandwidth for DES-MD5



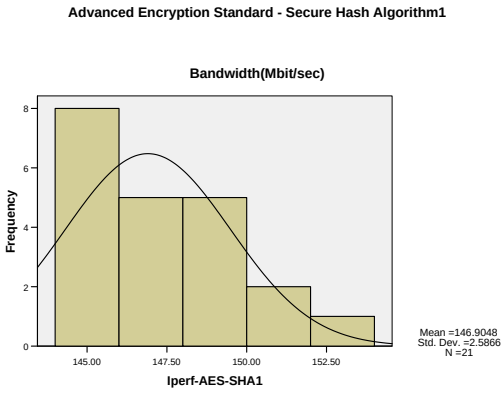


Figure 5.38 Bandwidth for AES-SHA **Figure 5.39 Bandwidth for AES-SHA1**

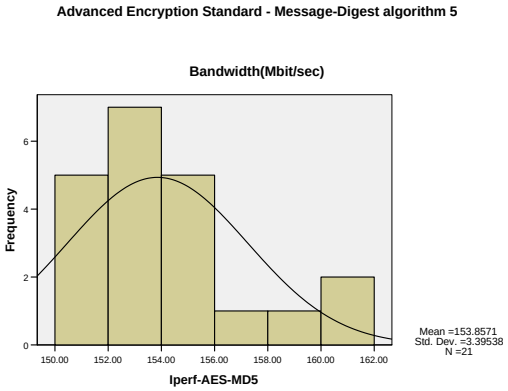


Figure 5.40 Bandwidth for AES-MD5

5.5.2 Statistical Analysis of Iperf Bandwidth Results

In this section the bandwidth results measured using Iperf for the different combinations will be compared statistically and analyzed.

Table 5.4 below shows statistical analysis of the results of Bandwidth measured by Iperf for different combinations.

Table 5.4 Statistical Analysis of Iperf Bandwidth Results

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
ipbfsha	48.13	49.70	42.80	1.97	3.86
ipbfsha1	47.96	50.30	43.20	1.58	2.49
ipbfmd5	42.01	50.20	32.10	5.68	32.28
ipdesha	49.12	49.50	48.50	.31	.10
ipdesha1	48.69	49.50	42.40	1.71	2.92
ipdemd5	50.04	50.70	44.50	1.37	1.87
ipaesha	134.24	140.00	130.00	2.62	6.89
ipaesha1	146.90	153.00	144.00	2.59	6.69
ipaemd5	153.86	162.00	150.00	3.40	11.53

Figure 5.41 below shows the bandwidth mean values for different combinations

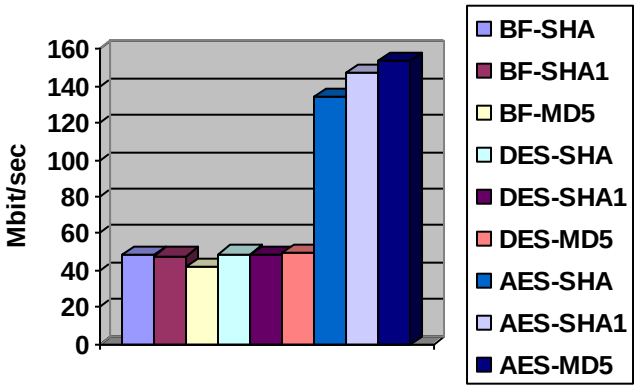


Figure 5.41 Bandwidth Mean values for Different Combinations

Again we notice that the AES-SHA1, AES-SHA and AES-MD5 have the highest throughput which is about three times the throughput of the other combinations. This may be explained by the testbed specifications, because they were carried out in the second testbed as explained in the previous sections.

A part from this we notice that Bandwidth isn't affected with the change in the combination applied, which is expected since Bandwidth should depend on the hardware, cables and network devices specifications not the data which passes through the network.

5.6 Jitter

Iperf program was used to measure the Bandwidth of the implemented SSL VPN, . The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, twenty times each. In the following sections the results for the different combinations will be given and analyzed.

5.6.1 Jitter Histograms for Different Combinations

In this section the jitter histograms for different combinations will be given.

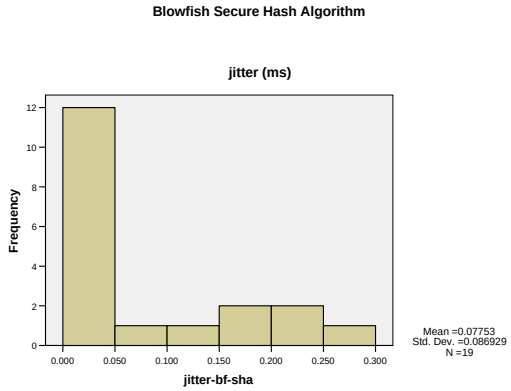


Figure 5.42 Jitter for BF-SHA

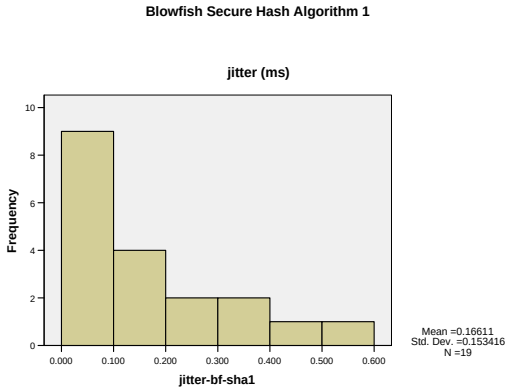


Figure 5.43 Jitter for BF-SHA1

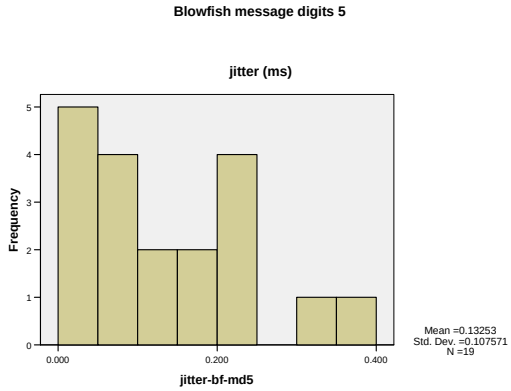


Figure 5.44 Jitter for BF-MD5

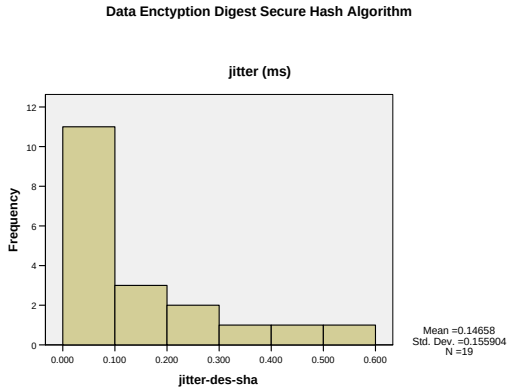


Figure 5.45 Jitter for DES-SHA

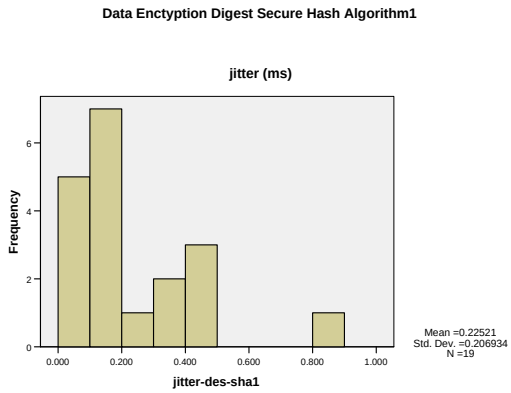


Figure 5.46 Jitter for DES-SHA1

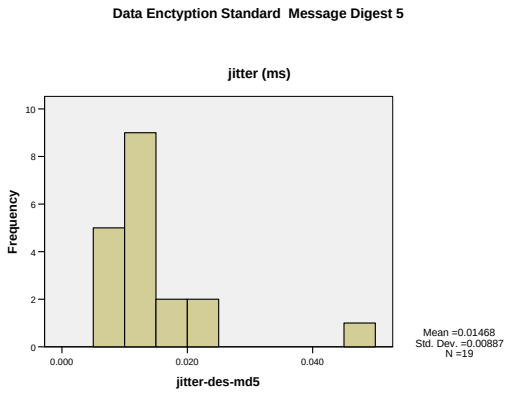


Figure 5.47 Jitter for DES-MD5

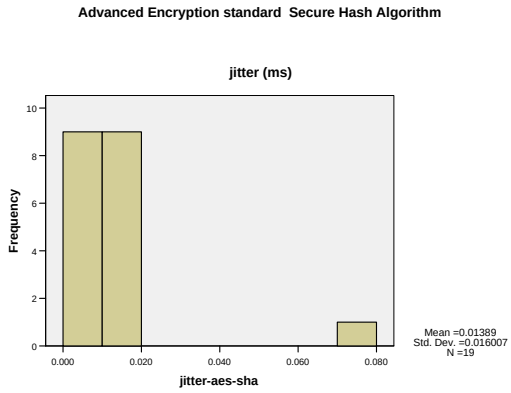


Figure 5.48 Jitter for AES-SHA

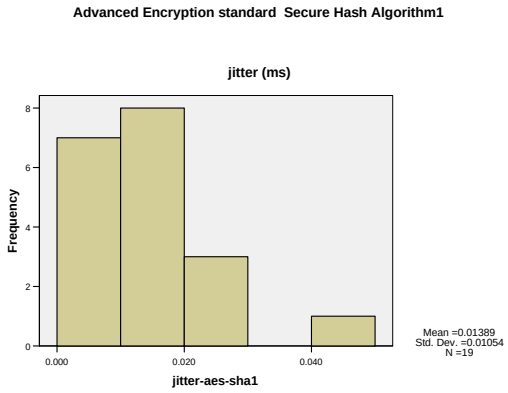


Figure 5.49 Jitter for AES-SHA1

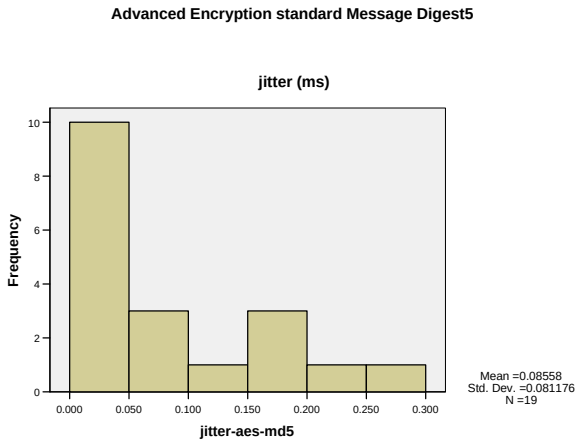


Figure 5.50 Jitter Histogram for AES-MD5

5.6.2 Statistical Analysis of Jitter Results

In this section the jitter results measured using Iperf for the different combinations will be compared statistically and analyzed.

Table 5.4 below shows statistical analysis of the results of jitter measured by Iperf for different combinations.

Table 5.5 Statistical Analysis of Jitter Results

Algorithim	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

In table 5.5, for each of the combinations the mean, maximum, minimum, standard deviation and variance were calculated.

Figure 5.51 below shows Jitter mean values for different combinations

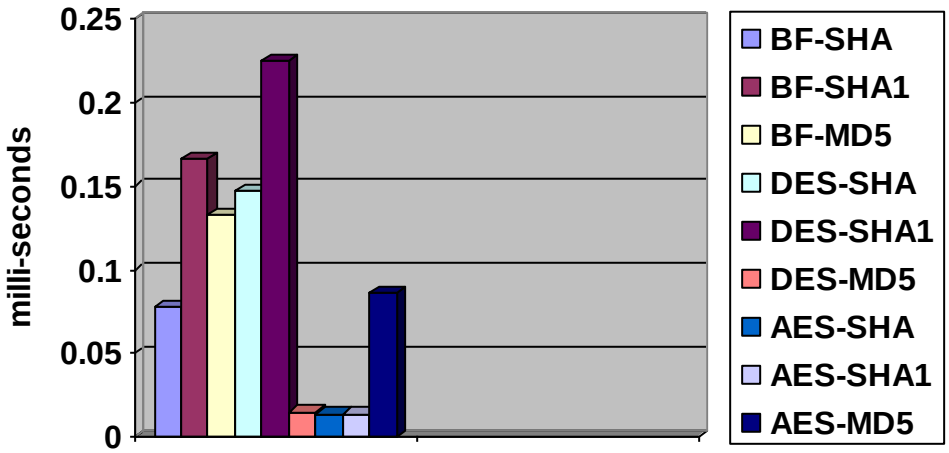


Figure 5.51 Jitter Mean Values for Different Combinations

As shown from the above figures jitter appears to have a random nature and could be assumed to have a guassian disrtibution, independent of the combination applied which can be explained by the two facts:

- It is caused, mainly, by thermal noise which have a Gaussian distribution
- The central limit theorem which states that composite effect of many uncorrelated noise sources, regardless of the distribution approaches a Gaussian distribution

5.7 Ftp Test

Ftp was configured in the implemented SSL VPN to measure the transfer rate.The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, five times each. In the following sections the results for the different combinations will be given and analyzed.

5.7.1 Transfer Rate Histograms for Different Combinations

In this section the transfer rate histograms for different combinations will be given.

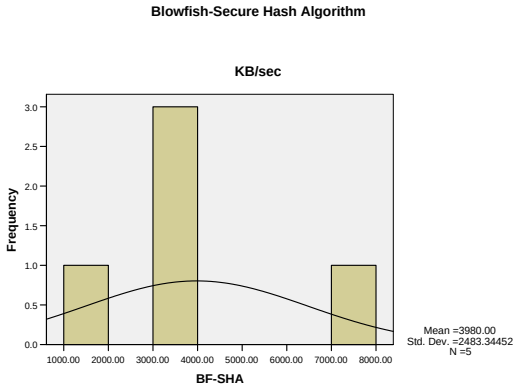


Figure 5.52 Transfer rate for BF-SHA

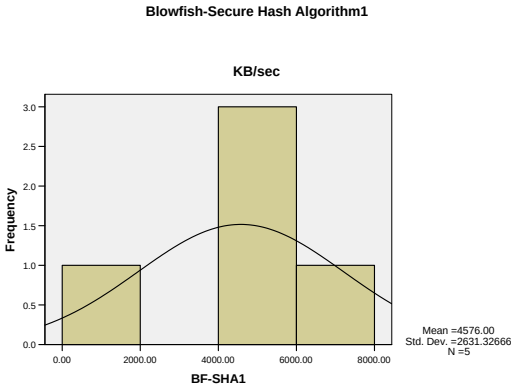


Figure 5.53 Transfer rate for BF-SHA1

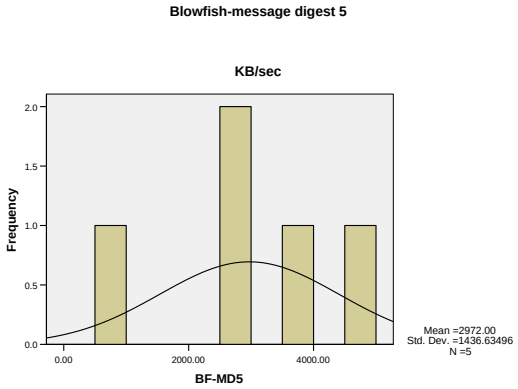


Figure 5.54 Transfer rate for BF-MD5

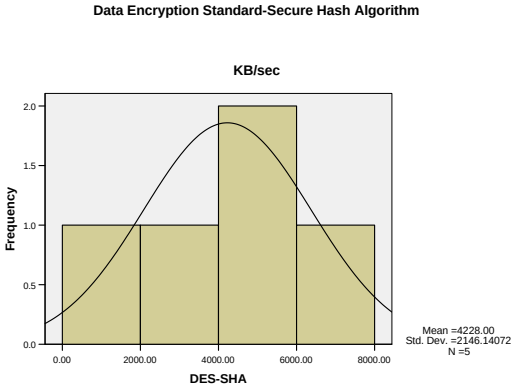


Figure 5.55 Transfer rate for DES-SHA

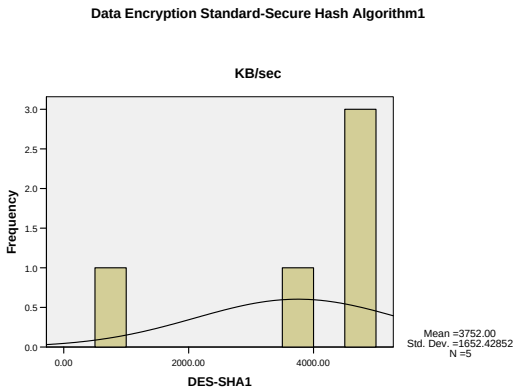


Figure 5.56 Transfer rate for DES-SHA1

Figure 5.57 Transfer rate for DES-MD5

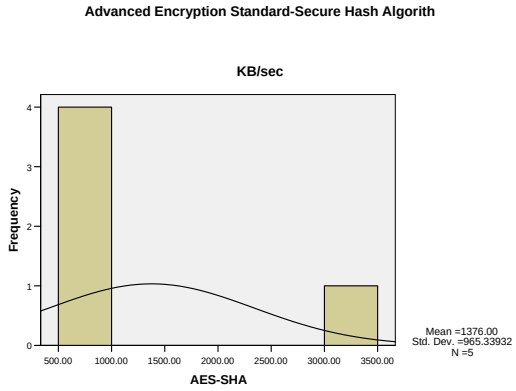


Figure 5.58Transfer rate for AES-SHA

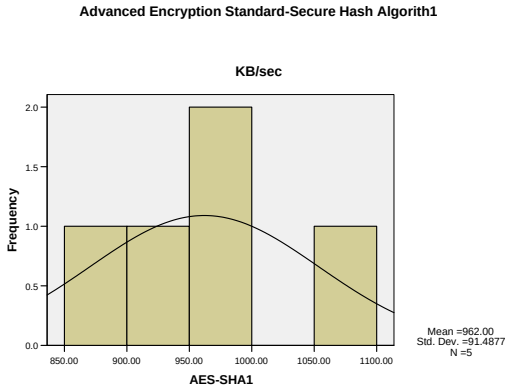


Figure 5.59Transfer rate for AES-SHA1

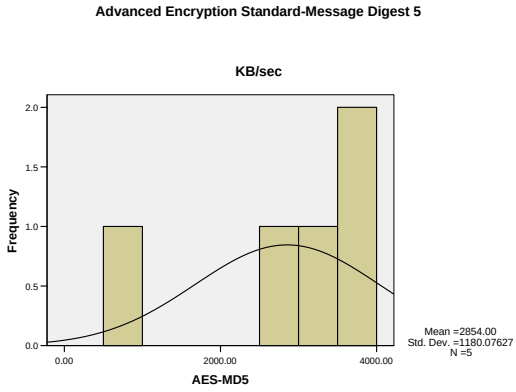


Figure 5.60 Transfer rate for AES-MD5

5.7.2 Statistical Analysis of Ftp Results

In this section the Transfer results measured using ftp for the different combinations will be compared statistically and analyzed.

Table 5.6 below shows statistical analysis of the results of transfer rate measured by ftp for different combinations.

Table 5.6 Statistical Analysis of Ftp Results

Algorithm	Mean	Maximum	Minimum	Standard Deviation	Variance
bfsha	3980.00	7900.00	1000.00	2483.34	6167000.00
bfsha1	4576.00	8000.00	880.00	2631.33	6923880.00
bfmd5	2972.00	4800.00	860.00	1436.63	2063920.00
dessha	4228.00	6100.00	940.00	2146.14	4605920.00
dessha1	3752.00	4800.00	860.00	1652.43	2730520.00
desmd5	3920.00	5100.00	1000.00	1684.34	2837000.00
aessha	1376.00	3100.00	850.00	965.34	931880.00
aessha1	962.00	1100.00	850.00	91.49	8370.00
aesmd5	2854.00	3800.00	870.00	1180.08	1392580.00

In table 5.6, for each of the combinations the mean, maximum, minimum, standard deviation and variance were calculated.
Figure 5.61 shows below the mean values for different combinations

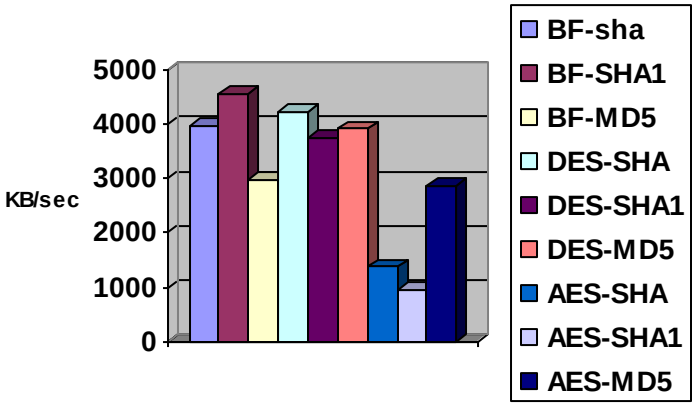


Figure 5.61 Transfer Rate Mean Values for Different Combinations

As figure 5.61 shows, AES-SHA, AES-SHA1 and AES-MD5 combinations have the least transfer rate which may be explained by the key size of AES cryptographic cipher which is larger than the key size used in blowfish and DES cryptographic ciphers.

Therefore AES needs more computational effort than blowfish and DES, thus it is acceptable that AES should have the least transfer rate.

Considering the three combinations of AES cipher we notice that the combination AES-SHA1 has the least transfer rate, this may be explained by the block size generated by SHA1 hash function (160 bits) which is larger than the block size generated by both SHA and MD5 hash algorithm (both are 128 bits).

5.8 NFS access time

NFS was configured in the implemented SSL VPN to measure access time. The test was done for each of the combinations of the cryptographic ciphers and hash algorithms, five times each. In the following sections the results for the different combinations will be given and analyzed.

5.8.1 Access time Histograms for Different Combinations

In this section histograms of the results for the different combinations will be given.

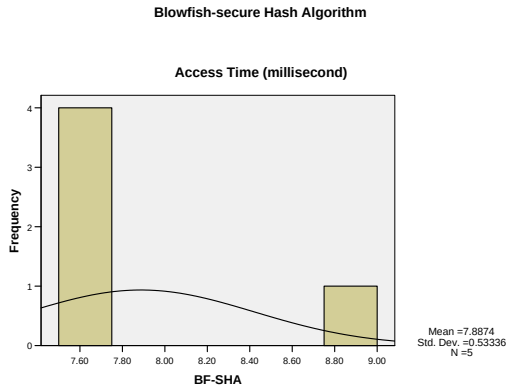


Figure 5.62 Access time for BF-SHA

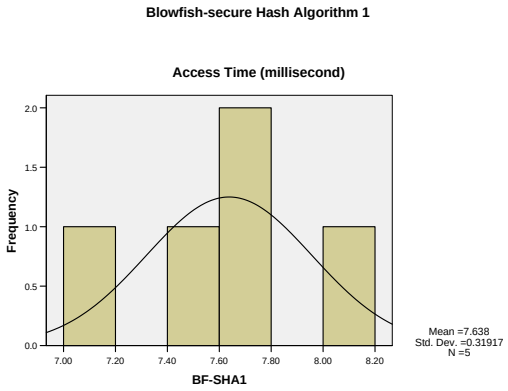


Figure 5.63 Access time for BF-SHA1

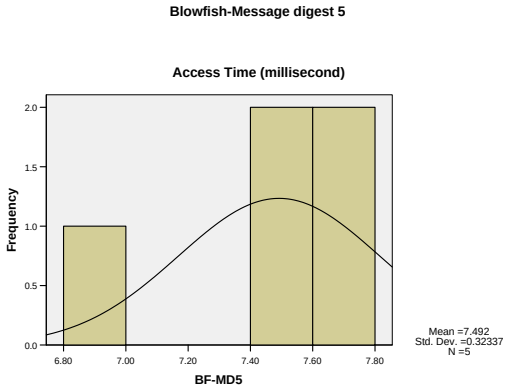


Figure 5.64 Access time for BF-MD5

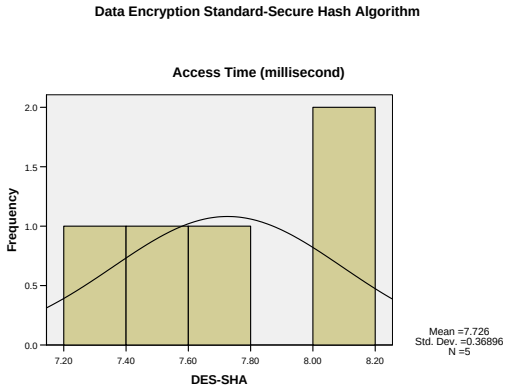


Figure 5.65 Access time for DES-SHA

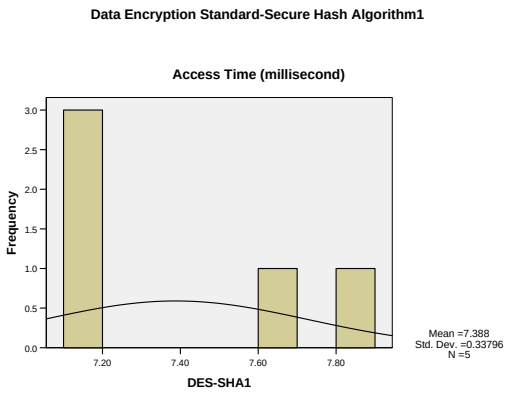


Figure 5.66 Access time for DES-SHA1

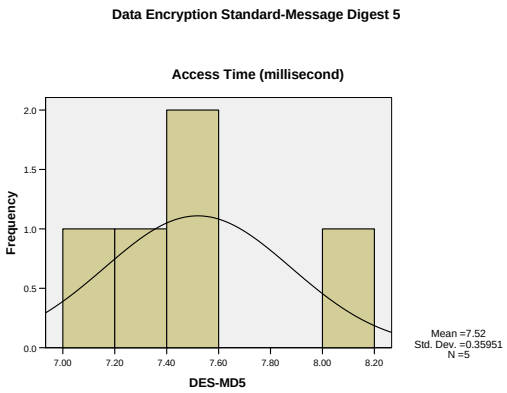


Figure 5.67 Access time for DES-MD5

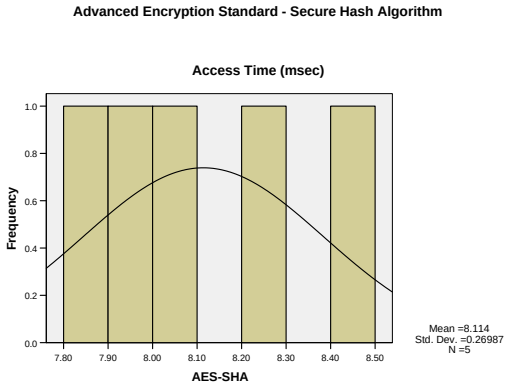


Figure 5.68 Access time for AES-SHA

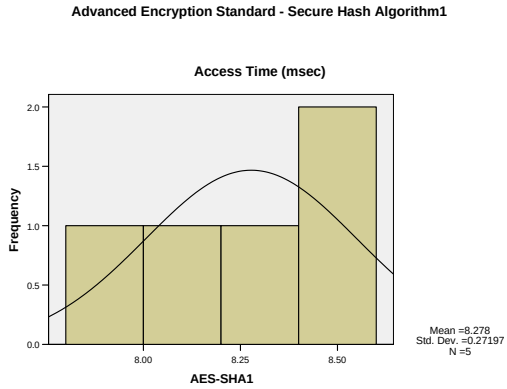


Figure 5.69 Access time for AES-SHA1

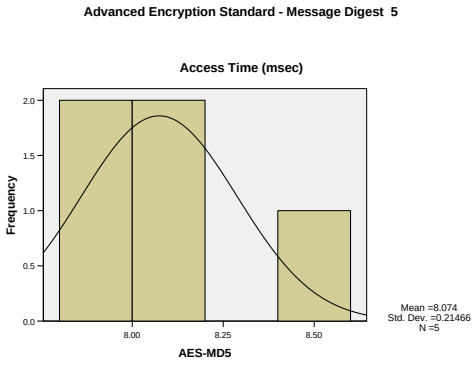


Figure 5.70 Access time for AES-MD5

5.8.2 Statistical Analysis of NFS Access Time Results

In this section the access time results measured using NFS and Tcpdump for the different combinations will be compared statistically and analyzed.

Table 5.7 shows statistical analysis of the results of packet access time measured by NFS and Tcpdump for different combinations.

Table 5.7 Statistical Analysis of NFS Access Time Results

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

In table 5.7, for each of the combinations the mean, maximum, minimum, standard deviation and variance were calculated.

Figure 5.71 below shows mean NFS access time for different combinations.

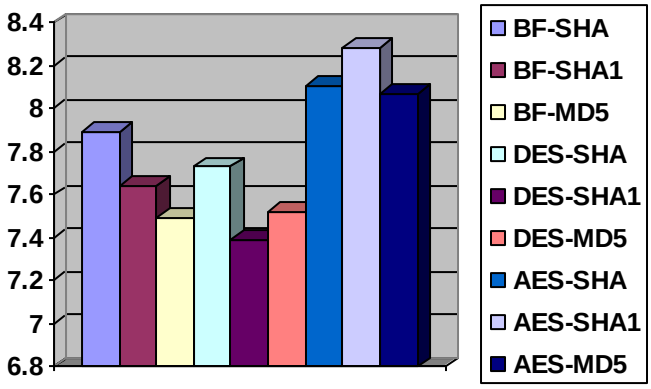


Figure 5.71 Access Time Mean Values for Different Combinations

As figure 5.71 shows, AES-SHA, AES-SHA1 and AES-MD5 combinations have the largest access time values which may be explained by the key size of AES cryptographic cipher which is larger than the key size used in blowfish and DES cryptographic ciphers. Therefore AES needs more computational effort than blowfish and DES, thus it is acceptable that AES should have the largest transfer rate.

Considering the three combinations of AES cipher we notice that the combination AES-SHA1 has the largest access time, this may be explained by the block size generated by SHA1 hash function (160 bits) which is larger than the block size generated by both SHA and MD5 hash algorithm (both are 128 bits).

5.9 IPSEC Results

In this section IPSEC performance evaluation results will be given and analyzed.

Round Trip Time (RTT) Histogram

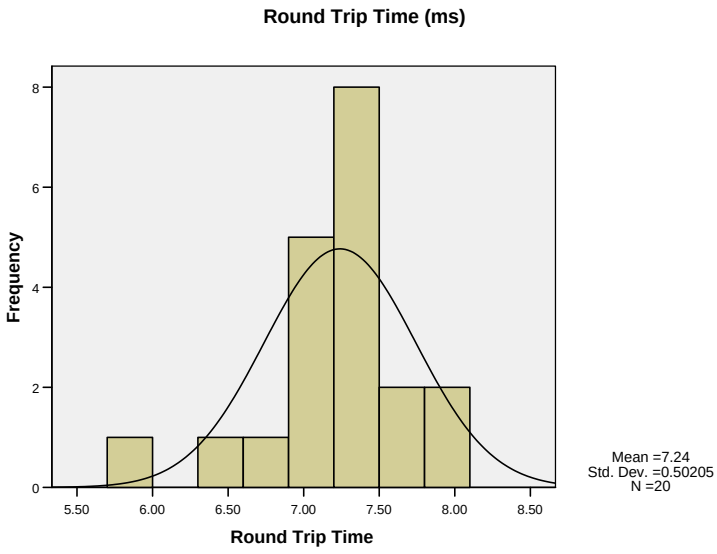


Figure 5.72 IPSEC RTT Test Results

Throughput measured using Jperf

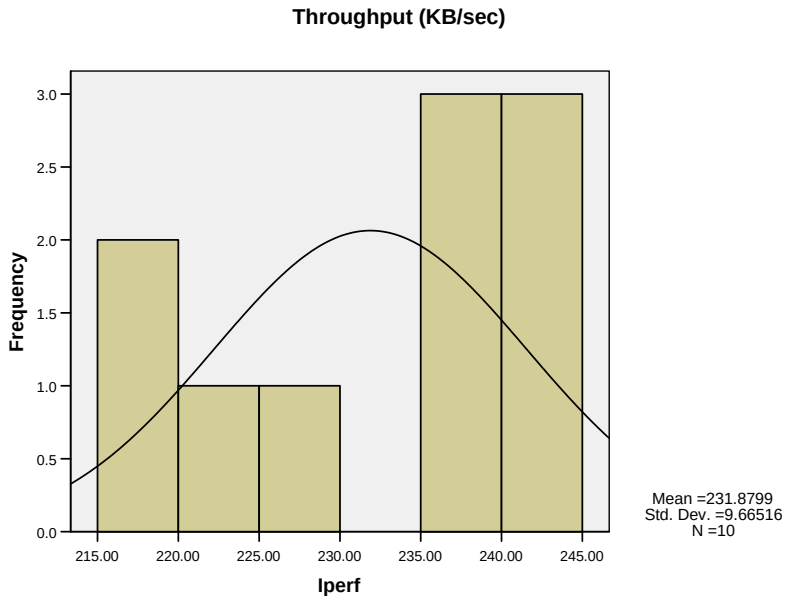


Figure 5.73 IPSEC Throughput Results

IPSEC results statistical analysis

Table 5.8 below shows statistical analysis of IPSEC results

Table 5.8 IPSEC Results Statistical Analysis

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

IPSEC tests were done in a hardware-based VPN; traffic passes through many routes with many hops and thus it encounters many delays such as propagation and transmission delays, whereas SSL VPN tests were done on two machines connected to the same device that's why a direct comparison isn't reasonable in this case.

6 Conclusion and Recommendations

6.1 Conclusion

In this project an SSL VPN has been successfully implemented. The SSL VPN was implemented using an Open-source, platform-independent with modular design solution which is considered a suitable option for researches because of the ability to evolve and minimum cost. A performance evaluation study was done for the implemented SSL VPN taking into consideration the effect of encryption and authentication techniques on the performance of the VPN. Three of the most used cryptographic ciphers and hash algorithms were chosen for the study. Network performance measures were tested on the implemented SSL VPN when applying each of the combinations of these cryptographic ciphers and hash algorithms. Network performance measures calculated are: round trip time (RTT), packet loss, bandwidth or throughput, transfer rate using FTP, access time using NFS and jitter.

The results of the study are was:

- Packet loss and RTT isn't affected with the combination applied
- Throughput and Bandwidth isn't affected with the combination applied
- Transfer rate decreases with the amount of encryption applied, as noted for cryptographic ciphers with larger key size the transfer rate is less than those with smaller key size.
- NFS access time increases with the amount of encryption applied, as noted for cryptographic ciphers with larger key size access time is larger than those with smaller key size.
- Jitter has a random nature

6.2 Recommendations

In this section ideas for future work that could be done to further enhance and extend the study of SSL VPN technology will be presented

Appliance-based SSL VPN Implementation and Performance evaluation

Implementation and performance evaluation of an Appliance-based SSL VPN is necessary since most companies, corporations and universities using this technology use an appliance-based SSL VPN.

Appliance-based SSL VPN vs. Software Products SSL VPN

A comparison between the two options of implementation of SSL VPN is very useful to be able to choose between them from a performance point of view.

Appliance-based SSL VPN vs. Appliance-based IPSEC VPN

Since IPSEC VPN is the most known type of VPN a direct comparison between appliance-based SSL VPN and appliance-based IPSEC VPN is essential, especially now, because many companies apply IPSEC VPN, such a study could provide a reference from which these companies and corporations could take a decision to switch to SSL VPN or keep their systems.

References

[1] Mike Erwin , Scott, Wolfe, *Virtual Private Networks*, O'Reilly, 2nd Edition, 1999

[2] James Henry Carmouche, *IPSEC Virtual Private Network Fundamentals*, Cisco Press, July 19, 2006

[3] Jon C. Snader, *VPNs Illustrated: Tunnels, VPNs, and IPsec***,** Addison Wesley Professional, October 26, 2005

[4] Behrouz A. Forouzan, *TCP/IP Protocol Suite,* TaTa Mc-graw-Hill, Third Edition, 2006

[5] Anon$_1$, *Virtual Private Networks*, URL: www.netsecurity.about.com, accessed on October, 2008

[6] Anon$_2$, *SSL VPN*, URL: http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib, accessed on October, 2008

[7] James L. Brown, *SSL VPNs,* URL:www.ittsg.ox.ac.uk, accessed on on October 9, 2008

[8] T. Braun, M. Günter, I. Khalil, L. Liu, *Performance Evaluation of Virtual Private Network,* Institut für Informatik und Angewandte Mathematik (IAM), University of Bern

[9] Shashank Khanvilkar , Ashfaq Khokhar, " Virtual Private Networks: An Overview with Performance Evaluation ", *IEEE Communications Magazine*, October 2004

[10] Charlie Hosner, *Openvpn and the SSL VPN revolution,* White paper, 2005

[11] Joseph Steinberg, Tim Speed, *SSL VPN : Understanding, evaluating and planning secure, web-based remote access*, first edition, Prentice Hall, 2007

[12] Anon$_3$, *Openvpn*, URL: http://en.wikipedia.org/wiki/OpenVPN, accessed on Feb 2008

[13] Joseph D. Sloan, *Network Troubleshooting Tools*, O'Reilly, 2nd Edition, August 2001

[14] José M. Caballero, "Dealing with Jitter Wander", *Trend Communications*, 2006

[15] Terry Collings, Kurt Wall, *Red Hat Linux Networking and System Adminstration*, Wiley, Third edition, 2006

[16] Joshua Erdman, *Switches vs. Hubs*, URL: http://www.networkclue.com/Get a Clue Switches vs_ Hubs.htm, accessed on June, 2008

[17] William Stallings, *Cryptography and Network Security Principles and Practices*, Prentice Hall, Fourth edition, 2005

Appendix A
OpenVPN Server Configuration File

```
port 1194

proto udp

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.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh "/etc/openvpn/dh1024.pem"

ifconfig 10.8.0.1 10.8.0.2
  ifconfig 10.8.0.1 255.255.255.0

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
```

```
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

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody
;daemon
# 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
```

Appendix B

OpenVPN Client Configuration File

```
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun
;keepalive 10 120
# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 172.16.15.106  1194

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody
;daemon
# Try to preserve some state across restarts.
persist-key
persist-tun
```

```
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca "/etc/openvpn/ca.crt"
cert "/etc/openvpn/client1.crt"
key "/etc/openvpn/client1.key"


# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher DES-EDE3-CBC
;cipher BF-CBC
cipher AES-256-CBC
auth SHA1
ifconfig 10.8.0.2 10.8.0.1
ifconfig 10.8.0.2 255.255.255.0
pull

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 5
```

Appendix C

Useful Files for FTP and NFS Configuration

This appendix presents some of the necessary files for configuration of both ftp and NFS.

C.1 exports file

This file describes the files to be exported by giving their path and the hosts allowed to access them and the permissions applied on the hosts.

```
/home/openvpn/Desktop/ab
10.8.0.6/255.255.255.255:(rw,async,no_root_squash)
```

C.2 hosts. allow file

This file describes the names of the hosts which are allowed to use the local INET services

```
ALL: LOCAL,10.8.0.1,10.8.0.6: allow
```

C.3 hosts. deny file

This file describes the names of the hosts which are not allowed to use the local INET services

```
ALL : ALL
```