CS 7530: Advanced Cryptography, Summer 2024, Instructor – Ahyoung Lee

Implementation and Optimization of IPsec Site-to-Site VPN in Cloud Environments

Sharon Edward John
Department of Computer Science
Kennesaw State University
Marietta, GA
sedwardj@students.kennesaw.edu

ABSTRACT

As organizations manage and utilize IT resources in the cloud, they benefit from the unparalleled scalability, flexibility and cost efficiency. The shift to cloud systems also brings forth security challenges, especially when there is communication between on-premises infrastructure and cloud-based resources. This research focuses on designing, implementing, and evaluating an IPsec-based site-to-site VPN solution within cloud environments to address these challenges. The primary objective is to establish a secure and efficient communication channel that mitigates the performance and security issues inherent in cloud infrastructures. This involves selecting appropriate encryption protocols, authentication mechanisms and key management protocols. IPsec is a well-established protocol for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet in a communication session. IPsec provides a high level of security, making it a suitable choice for establishing secure tunnels over the Internet. Site-to-site IPsec VPNs extend this security capability, allowing multiple sites or networks to connect securely over a public network, such as the Internet.

KEYWORDS

Cloud security, Internet Protocol Security (IPSec), Advanced Encryption Standard (AES-256), Virtual Private Network (VPN).

1. INTRODUCTION

Cloud computing has become an integral part of modern IT infrastructure, providing numerous advantages such as scalability, flexibility, and cost savings. However, the security of data transmitted between on-premises networks and cloud resources is a critical concern. IPsec (Internet Protocol Security) offers a robust framework for securing data in transit, making it a suitable choice for establishing secure site-to-site VPNs in cloud environments.

Secure communication between on-premises infrastructure and cloud servers is important as many organizations have adopted cloud services. In cloud, access is provided to customers, contractors, suppliers and developers. Hence security is a primary concern for organizations, and the users of cloud distrust the public cloud devices. Since different products of the cloud infrastructure is managed independently, delivering and maintaining security across multiple devices can be challenging.

IPsec provides a secure encryption framework to protect the data. And IPsec VPNs restrict the cloud based server from being able to access a private network while allowing the network to connect to the services of the public cloud server. This enhances the trust in cloud services.

There are some challenges that need to be addressed while implementing a secure VPN tunnel in the cloud. Configuring and managing IPsec VPNs across diverse cloud environments can be complex due to variations in cloud provider architectures and services. Ensuring compatibility and interoperability between on-premises infrastructure and cloud platforms requires good planning and expertise. Security is another critical challenge, as maintaining consistent security policies and managing cryptographic keys across multiple devices can be difficult. Performance issues, such as latency and bandwidth limitations, can arise due to the overhead of encryption and decryption processes, impacting the efficiency of data transmission. Additionally, scalability can be problematic, as the VPN must be able to handle fluctuating workloads and growing data traffic without compromising security or performance. Troubleshooting and maintaining the VPN can be demanding, requiring continuous monitoring and timely resolution of issues to ensure reliable and secure connectivity. The implementation will focus on ensuring that the VPN is has minimal latency and is interoperable, while maintaining high levels of security.

Implementing a site-to-site VPN solution delivers a highly secure means to maintain, control and access to corporate assets in the cloud computing environments. Implementing an IPsec site-to-site VPN in the cloud also leverages the cost benefits of using cloud while maintaining high levels of security, reducing the need for expensive dedicated connections.

This paper explores the challenges and solutions associated with implementing and optimizing IPsec site-to-site VPNs in cloud environments. The focus is on ensuring secure, efficient communication while addressing performance and security challenges unique to cloud infrastructures.

2. LITERATURE REVIEW

IPSec is recognized for its security features and performance. IPSec VPNs are used to secure internet communication by encrypting and authenticating data. An IPSec protocol is implemented using Strongswan software tool and the performance of IPSec tunnel on IPv6 based on throughput using various cryptographic algorithms was measured. Throughput is an important factor in measuring the performance of a network. IPSec performance was best on an AES-128 algorithm in the GCM mode. [7]

Encryption algorithms are crucial in ensuring the confidentiality, integrity, and authenticity of the data. The choice of encryption algorithm can impact the performance and security of the VPN. Another study on the encryption algorithms on an IPSec site-to-site shows AES-256 encryption is best suited for the tunnel implementation due to its strong security. [3] A site-to-site VPN tunnel gives the end users a dedicated network for communication while using a shared infrastructure. Site-to-Site VPNs connect entire networks to each other, enabling secure communication between different locations. Site-to-Site IPSec VPNs are introduced as a solution for establishing secure links between different geographical locations.

The paper [1] explains key components of IPSec, including Security Associations (SAs), the Authentication Header (AH), and the Encapsulating Security Payload (ESP). The role of Internet Key Exchange (IKE) in negotiating and establishing SAs is discussed. The authors provide a step-by-step guide for configuring a Site-to-Site IPSec VPN tunnel between routers. Steps include defining the VPN tunnel, configuring IKE policies, setting up encryption and authentication protocols, and establishing SAs. A practical case study is presented where a Site-to-Site IPSec VPN tunnel is implemented between two routers in a real-world scenario. Configuration examples using router command-line interface (CLI) commands are provided to illustrate the process.

The authors of [8] have implemented an IPSec VPN for inter-cloud services that copes with an increasing number of network users, traffic, changing networks and minimizing delay. This VPN tunnel is based on the hub-and-spoke VPN architecture.

In this project a VPN tunnel will be created using IPSec protocol on a cloud setup. A secure channel will be setup between two cloud servers and the communication packets will be encrypted using AES-

256 encryption. This project aims to establish a secure and efficient communication channel between two virtual machines (VMs) in a cloud environment using an IPsec-based site-to-site VPN. In addition to the encryption, the packets will be compressed to improve the throughput in the network.

3. PROPOSED SOLUTION

An IPSec tunnel will be setup between two cloud based VMs. The client will send messages to the server, with each message encrypted using AES encryption. The Encapsulating Security Payload (ESP) header in IPsec will provide additional encryption for the IP packets. To optimize network performance, the encrypted packets will be compressed before transmission. Upon receiving the packets, the server will decompress and decrypt them.

The use of AES encryption and IPsec ESP ensures that data is protected during transmission, providing security against eavesdropping and tampering. Compression of encrypted packets reduces their size, improving network performance by minimizing latency and increasing throughput. The solution can be scaled to accommodate additional VMs or higher data volumes by configuring additional IPsec tunnels and optimizing compression algorithms. The IPsec-based solution is compatible with various cloud providers and can be easily integrated into existing cloud infrastructures.

Fig. 1 shows the architecture of the site-to-site setup. The VPN is established within the cloud server and a subnet is created under the server. The VMs are created within the subnets and the data is sent from the VMs.

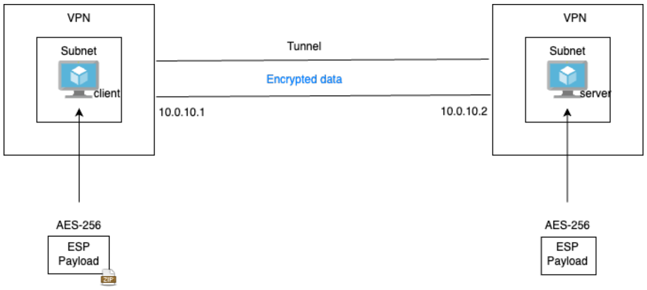


Figure 1: Site-to-site VPN architecture

4. IMPLEMENTATION

4.1. System Design

Cloud server: Microsoft Azure Cloud

Operating System: Linux Ubuntu 20.04.6 LTS

Packet Analyzer: Wireshark

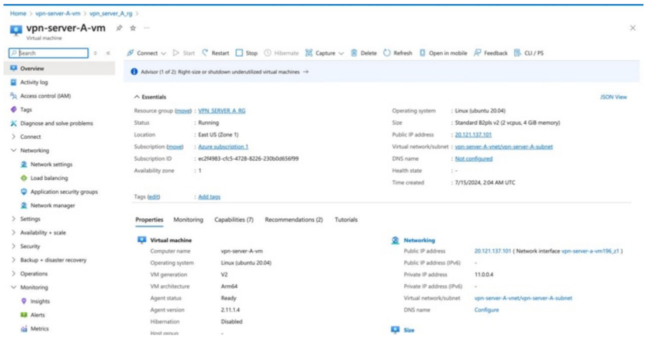
Compression software: GZIP

Programming language: Python 3.8

4.2. Cloud Setup

A VM is deployed on the client and server side in the Azure Cloud and Virtual Private Network is setup in between the VMs. The network is created in the address space 11.0.0.0/26 and 12.0.0.0/26. A subnet is created under the virtual network at 11.0.0.0/28.

Create a security group under the subnet to allow traffic from the client along with the local ssh access rule. The Linux server is created under the subnet and a 2-core machine is deployed with an assigned IP of 11.0.0.4 and 12.0.0.4.



vpn-server-A-vm
Virtual machine

Advisor (1 of 2): Right-size or shutdown underutilized virtual machines →

Essentials

Resource group (move) : VPN_SERVER_A_RG
Status : Running
Location : East US (Zone 1)
Subscription (move) : Azure subscription 1
Subscription ID : ec2f4983-cfc5-4728-8226-230b0d656f99
Availability zone : 1
Tags (edit) : Add tags

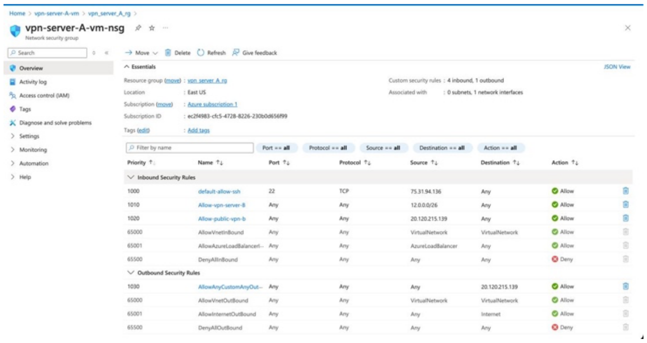
Operating system : Linux (ubuntu 20.04)
Size : Standard B2pls v2 (2 vcpus, 4 GiB memory)
Public IP address : 20.121.137.101
Virtual network/subnet : vpn-server-A-vnet/vpn-server-A-subnet
DNS name : Not configured
Health state : -
Time created : 7/15/2024, 2:04 AM UTC

Virtual machine
Computer name vpn-server-A-vm
Operating system Linux (ubuntu 20.04)
VM generation V2
VM architecture Arm64
Agent status Ready
Agent version 2.11.1.4
Hibernation Disabled
Host group -

Networking
Public IP address 20.121.137.101 (Network interface vpn-server-a-vm196_z1)
Public IP address (IPv6) -
Private IP address 11.0.0.4
Private IP address (IPv6) -
Virtual network/subnet vpn-server-A-vnet/vpn-server-A-subnet
DNS name Configure

Size

Figure 2: VPN server setup



Home > vpn-server-A-vm > vpn_server_A_rg >

vpn-server-A-vm-nsg
Network security group

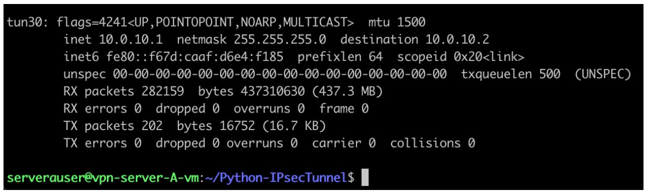
Resource group (move) : vpn server A rg
Location : East US
Subscription (move) : Azure subscription 1
Subscription ID : ec2f4983-cfc5-4728-8226-230b0d656f99
Tags (edit) : Add tags
Custom security rules : 4 inbound, 1 outbound
Associated with : 0 subnets, 1 network interfaces

Priority	Name	Port	Protocol	Source	Destination	Action
Inbound Security Rules						
1000	default-allow-ssh	22	TCP	75.31.94.136	Any	Allow
1010	Allow-vpn-server-B	Any	Any	12.0.0.0/26	Any	Allow
1020	Allow-public-vpn-b	Any	Any	20.120.215.139	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerI...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
Outbound Security Rules						
1030	AllowAnyCustomAnyOut...	Any	Any	Any	20.120.215.139	Allow
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

Figure 3: Configuring security rules

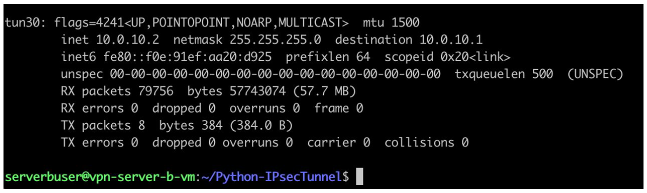
4.3. Tunnel setup

A tunnel interface has been setup on both servers and the message sent from the client is encapsulated in packets and sent through the tunnel.



```
tun30: flags=4241<UP,POINTOPOINT,NOARP,MULTICAST>  mtu 1500
        inet 10.0.10.1  netmask 255.255.255.0  destination 10.0.10.2
        inet6 fe80::f67d:caaf:d6e4:f185  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 282159  bytes 437310630 (437.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 202  bytes 16752 (16.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

serverauser@vpn-server-A-vm:~/Python-IPsecTunnel$ 
```

Figure 4: Tunnel Setup on sender side



```
tun30: flags=4241<UP,POINTOPOINT,NOARP,MULTICAST>  mtu 1500
        inet 10.0.10.2  netmask 255.255.255.0  destination 10.0.10.1
        inet6 fe80::f0e:91ef:aa20:d925  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 79756  bytes 57743074 (57.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 384 (384.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

serverbuser@vpn-server-b-vm:~/Python-IPsecTunnel$ 
```

Figure 5: Tunnel Setup on receiver side

The peer IP of the two hosts were used to setup a socket connection, through which the encrypted packets are transferred and would be transmitted through the tunnel interface for other hosts.

4.4. Encryption

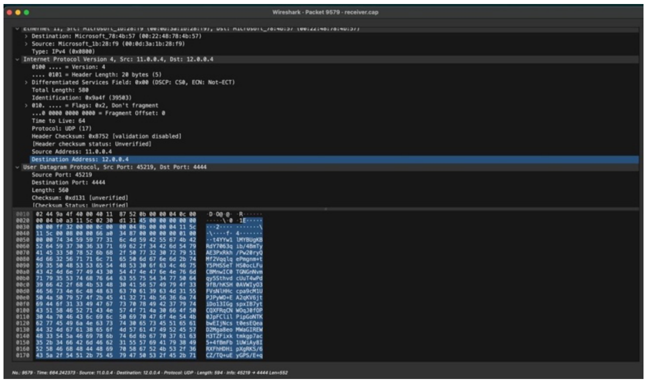
The payload is encrypted using the AES-256 key and is encapsulated into an ESP header. The ESP header along with the IP header and UDP header would be transmitted over the IPSec tunnel. The payload is compressed using GZip to reduce the response time in the communication.

4.5. Decryption

The compressed encrypted payload is extracted from the IP Packet and is decrypted using the AES algorithm and the decrypted data is transferred to the tunnel data for internal clients to access. The VM on the receiver side securely receives the message from the sender.

5. EVALUATION AND RESULTS

A secure tunnel has been established and the data is encrypted using the efficient AES-256 algorithm. To monitor the packets Wireshark packet analyzer has been used.



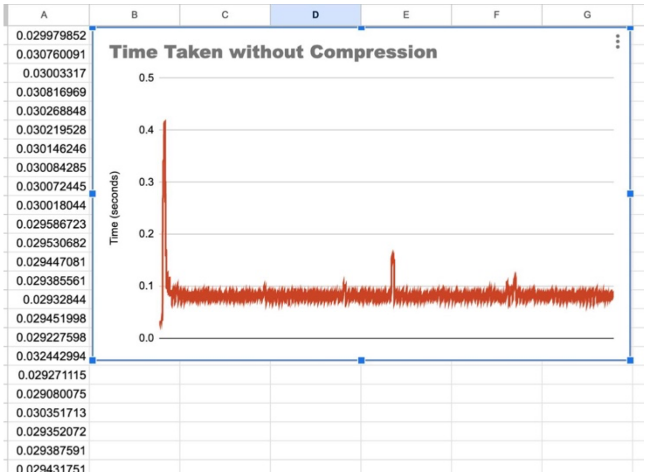
Wireshark · Packet 9579 · receiver.cap

```
v Ethernet II, Src: Microsoft_1b:28:f9 (00:0d:3a:1b:28:f9), Dst: Microsoft_78:4b:57 (00:22:48:78:4b:57)
   > Destination: Microsoft_78:4b:57 (00:22:48:78:4b:57)
   > Source: Microsoft_1b:28:f9 (00:0d:3a:1b:28:f9)
     Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 11.0.0.4, Dst: 12.0.0.4
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 580
     Identification: 0x9a4f (39503)
   > 010. .... = Flags: 0x2, Don't fragment
     ...0 0000 0000 0000 = Fragment Offset: 0
     Time to Live: 64
     Protocol: UDP (17)
     Header Checksum: 0x8752 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 11.0.0.4
     Destination Address: 12.0.0.4
v User Datagram Protocol, Src Port: 45219, Dst Port: 4444
     Source Port: 45219
     Destination Port: 4444
     Length: 560
     Checksum: 0xd131 [unverified]
     [Checksum Status: Unverified]

0010  02 44 9a 4f 40 00 40 11  87 52 0b 00 00 04 0c 00   ·D·O@·@· ·R······
0020  00 04 b0 a3 11 5c 02 30  d1 31 45 00 00 00 00 00   ·····\·0 ·1E·····
0030  00 00 ff 32 00 00 0c 00  00 04 0b 00 00 04 11 5c   ···2···· ·······\
0040  11 5c 00 08 00 00 66 a0  34 87 00 00 00 00 01 00   ·\····f· 4·······
0050  00 00 74 34 59 59 77 31  6c 4d 59 42 55 67 4b 42   ··t4YYw1 lMYBUgKB
0060  52 64 59 37 30 36 33 71  69 62 2f 34 42 6d 54 79   RdY7063q ib/4BmTy
0070  41 45 33 50 78 52 6b 68  2f 50 77 32 30 72 79 51   AE3PxRkh /Pw20ryQ
0080  4d 66 32 56 71 71 6c 71  65 50 6d 67 6e 6d 2b 74   Mf2Vqqlq ePmgnm+t
0090  59 35 50 48 53 53 65 54  48 53 30 6f 63 4c 46 75   Y5PHSSeT HS0ocLFu
00a0  43 42 4d 6e 77 49 43 30  54 47 4e 47 6e 4e 76 6d   CBMnwIC0 TGNGnNvm
00b0  71 79 35 53 74 68 76 64  63 55 75 54 34 77 50 64   qy5Sthvd cUuT4wPd
00c0  39 66 42 2f 68 4b 53 48  30 41 56 57 49 79 4f 33   9fB/hKSH 0AVWIyO3
00d0  46 56 73 4e 6c 48 48 63  63 70 61 39 63 4d 31 55   FVsNlHHc cpa9cM1U
00e0  50 4a 50 79 57 4f 2b 45  41 32 71 4b 56 36 6a 74   PJPyWO+E A2qKV6jt
00f0  69 44 6f 31 33 49 47 67  73 70 78 49 42 37 79 74   iDo13IGg spxIB7yt
0100  43 51 58 46 52 71 43 4e  57 4f 71 4a 30 66 4f 50   CQXFRqCN WOqJ0fOP
0110  30 4a 70 46 43 6c 69 6c  50 69 70 47 6f 4e 54 4b   0JpFClil PipGoNTK
0120  62 77 45 49 6a 4e 63 73  74 30 65 73 45 51 65 61   bwEIjNcs t0esEQea
0130  44 32 4d 67 61 38 65 6f  4d 57 61 47 49 52 45 57   D2Mga8eo MWaGIREW
0140  48 33 54 5a 46 69 78 6b  74 6d 6b 67 70 37 61 63   H3TZFixk tmkgp7ac
0150  35 2b 34 66 42 6d 46 62  31 55 57 69 41 79 38 49   5+4fBmFb 1UWiAy8I
0160  52 58 46 68 48 44 48 69  70 58 67 52 4b 53 2f 36   RXFhHDHi pXgRKS/6
0170  43 5a 2f 54 51 2b 75 45  79 47 50 53 2f 45 2b 71   CZ/TQ+uE yGPS/E+q

No.: 9579 · Time: 664.242373 · Source: 11.0.0.4 · Destination: 12.0.0.4 · Protocol: UDP · Length: 594 · Info: 45219 → 4444 Len=552
```

Figure 6: Packet trace between hosts

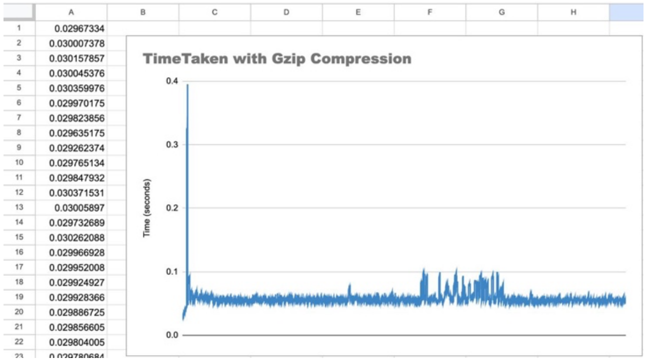
The start time has been encapsulated along with the message on the IP packet and extracted on the receiver side from which the total time was calculated. These data points are plotted in a graph below for comparison. The comparison shows reduction in the time taken for the packets to reach the receiver, when the packets are compressed.

The graph in *Fig.7* shows an average of 0.1 seconds for the packets to reach the receiver server.



A	B	C	D	E	F	G
0.029979852						
0.030760091						
0.03003317						
0.030816969						
0.030268848						
0.030219528						
0.030146246						
0.030084285						
0.030072445						
0.030018044						
0.029586723						
0.029530682						
0.029447081						
0.029385561						
0.02932844						
0.029451998						
0.029227598						
0.032442994						
0.029271115						
0.029080075						
0.030351713						
0.029352072						
0.029387591						
0.029431751						

Figure 7: Time to send data between hosts with uncompressed packets



	A	B	C	D	E	F	G	H
1	0.02967334							
2	0.030007378							
3	0.030157857							
4	0.030045376							
5	0.030359976							
6	0.029970175							
7	0.029823856							
8	0.029635175							
9	0.029262374							
10	0.029765134							
11	0.029847932							
12	0.030371531							
13	0.03005897							
14	0.029732689							
15	0.030262088							
16	0.029966928							
17	0.029952008							
18	0.029924927							
19	0.029928366							
20	0.029886725							
21	0.029856605							
22	0.029804005							
23	0.029780684							

Figure 8: Time to send data between hosts with compressed packets

The graph in *Fig.8* shows a reduction in time by an average of 0.05 seconds for the packets to reach the receiver server.

6. CONCLUSION:

The proposed IPsec site-to-site VPN solution provides a secure and efficient method for transmitting data between two VMs in a cloud environment. A secure encryption path has been setup between the hosts so that the communication is safe. By leveraging AES encryption, IPsec ESP, and packet compression, the solution ensures data security while optimizing network performance, making it a valuable approach for securing cloud-based communications.

REFERENCES

[1] Khaing, Ei & Than Nyunt, Khin & Moe, Sandar & Khaing, Mya. (2021). Implementation of Site to Site IPsec VPN Tunnel between Routers. International Journal of Scientific Research in Science, Engineering and Technology. 163-169. 10.32628/IJSRSET218133.

[2] Rathore, M.S., Razzaq, A., Hidell, M., & Sjödin, P. (2009). Site-to-Site VPN Technologies : A Survey.

[3] M. H. M. Zaharuddin, R. A. Rahman and M. Kassim, "Technical comparison analysis of encryption algorithm on site-to-site IPSec VPN," *2010 International Conference on Computer Applications and Industrial Electronics*, Kuala Lumpur, Malaysia, 2010, pp. 641-645, doi: 10.1109/ICCAIE.2010.5735013.

[4] Udayakumar, R.. "Deploying Site-To-Site VPN Connectivity: MPLS Vs IPSec." (2014).

[5] Hashiyana, Valerianus & Haiduwa, Titus & Suresh, Nalina & Bratha, Aubrey & Ouma, Flavia. (2020). Design and Implementation of an IPSec Virtual Private Network: A Case Study at the University of Namibia.

[6] William Stallings. 2002. Cryptography and Network Security: Principles and Practice (3rd. ed.). Pearson Education.

[7] P. Thiruvasagam, K. J. George, S. Arumugam and A. R. Prasad, "IPSec: Performance Analysis in IPv4 and IPv6," in *Journal of ICT Standardization*, vol. 7, no. 1, pp. 61-80, 2019, doi: 10.13052/jicts2245-800X.714