IPSec: Performance Analysis in IPv4 and IPv6

Prabhu Thiruvasagam and K. Jijo George

NEC India Private Limited, India
E-mail: prabhu.t@necindia.in; k.george@necindia.in

Received 02 December 2018;
Accepted 06 January 2019

Abstract

Internet Protocol security (IPSec) is an end-to-end security scheme to provide security at the IP network layer, but this comes with performance implications leading to throughput reduction and resource consumption. In this paper we present a throughput performance analysis of IPSec protocol, for both IPv4 and IPv6, using various cryptographic algorithms as recommended in the standards [13]. In this study we have considered only throughput performance for authenticated encryption algorithms AES-GCM and AES-CCM, encryption algorithms AES-CBC, AES-CTR, and 3DES, and authentication algorithms SHA1, SHA2 and XCBC. The result shows that AES-GCM provides better performance compared to the other recommended algorithms.

Keywords: IPSec, Authentication, 3GPP, NDS/IP, LTE/SAE, AES, IPv6.

1 Introduction

Initially, Internet Protocol (IP) was designed along with TCP, UDP and other protocols to communicate over the common internet medium without the aim of providing security services inherently. Due to the lack of security features

Journal of ICT, Vol. 7_1, 59–76. *River Publishers*
doi: 10.13052/jicts2245-800X.714

in internet protocols, communication over the internet is subject to various security threats. To address these security threats, IPSec standard was defined by IETF. IPSec architecture [1] is a suite of protocols providing the set of IP extensions for implementing security in the network (IP) layer for both IPv4 and IPv6. IPSec supports two security protocols: (1) Authentication Header (AH) and (2) Encapsulating Security Payload (ESP). AH provides integrity, data origin authentication and anti-replay (anti-playback) security features while ESP provides all the security services provided by AH and confidentiality. IPSec protocol supports two modes of operation, these are tunnel mode and transport mode. In tunnel mode operation, protection is provided for the entire IP packet. In transport mode operation, protection is provided only to the packet payload. In IPSec, security services are provided through the combination of cryptographic algorithms and security protocols. IPSec standard provides the architecture to setup the secure IP tunnel and allows users to choose cryptographic algorithms independently based on their requirements.

In IPSec implementations, two kinds of protocol suites are used to successfully establish the secure tunnels and protect the data that are transmitted in between the communicating entities. First one is Internet Key Exchange (IKE) [2, 3] and its associated protocols (ISAKMP, OAKLEY) [4, 5], which are used for authenticating the communicating entities, and establish the Security Associations (SA) between them. It offers flexible authentication and effective key negotiation methods, which makes it suitable for scalable deployments. Another one is IPSec security protocols (AH and ESP) [6, 7], which is used to protect the data transmitted over the network. IPSec is also employed by other internet protocols like Mobile IP, etc. [8].

Use of IPSec can secure the communication over the unsecured medium, but it consumes bandwidth and requires resources for performing the cryptographic function operations which may impact the performance based on the algorithms and security protocols it uses. As the technology evolves, existing cryptographic algorithms are being attacked or proved to be weak and new cryptographic algorithms are being proposed to secure the communication. Today's strong cryptographic algorithms may not be secure in the future, so time to time the recommendations for using cryptographic algorithms in IPSec framework vary.

Initially, IPv4 was designed to assign IP addresses to connected devices in the network. IPv4 address is 32 bits long and its range (2^32) is in millions which is lesser than the total population of the world. So, IPv6 [9, 10] was designed to meet the emerging needs due to the proliferation

of information technology and the number of connected computing devices. IPv6 uses an address length of 128 bits. The prediction of presence of millions to billions of internet of things (IoT) connected devices makes IPv4 migration to IPv6 essential. In the literature, we found that the performance of IPSec in IPv6 based network environment is not analysed sufficiently.

Performance analysis of IPSec has been done in the past using 100 Mbps network card, but most of the analysed cryptographic algorithms are outdated and proved to be weak [11, 12] and mostly performed in IPv4 based network environment.

In this experimental work, our contributions are:

- performance analysis of various recently recommended authenticated encryption associated data (AEAD) cryptographic algorithms for implementing IPSec by standards [13, 14]
- performance analysis with the focus on telecom core network using 1Gbps network interface card
- performance analysis of IPSec and associated cryptographic algorithms for IPv4 and IPv6
- evaluated the characteristics of TCP and UDP packets throughput, jitter and packet loss in IPSec enabled network.

We use Strongswan an open source software to establish SAs and protect data using IPSec protocols. From research studies [15–17] it is clear that ESP protocol along with tunnel mode is widely used and also recommended by standards to provide security services at the IP layer. Thus, in our experimental setup we use ESP protocol along with tunnel mode to analyse the performance of IPSec in host to host (gateway to gateway) scenario.

The organization of this paper is as follows. Section 2 discusses the IPSec performance analysis related works and Section 3 explains about IPSec standards and recommendations for implementing IPSec. Section 4 describes the experimental setup and Section 5 discusses the results and analysis of this work. Section 6 concludes the paper.

2 Related Work

There are research works on performance and overhead analysis of IPSec and associated cryptographic algorithms based on IPSec standards in IPv4 based network environment, but the analysed cryptographic algorithms are outdated now and proven to be weak. Particularly, the early analysis mostly focused on DES, 3DES, MD5 and SHA-1 cryptographic algorithms in IPSec framework and did not cover the performance analysis of AEAD algorithms

and other authentication algorithms recommended in the standards recently. For instance, O. Elkeelany *et al.* [18] and G. Hadjichristofi *et al.* [19] analysed the IPSec protocol performance using DES, MD5 and SHA-1 cryptographic algorithms and its overhead. C. Xenakis *et al.* [20] analysed the generic characterisation of the overheads imposed by IPSec and associated cryptographic algorithms in mobile devices in wireless environment (UMTS network), they analysed the overheads imposed by cryptographic algorithms (DES, MD5 and SHA-1). C. Shue *et al.* [21, 22] analysed the overhead imposed by cryptographic functions (3DES192, AES128, and AES256) in IPSec Processing and they compared the overhead caused by IKE protocols and IPSec protocols, and reported that cryptographic operations incur 32–60% of the total overhead in IPSec. Also they proposed cache resumption method to reduce the overhead in multi-client environment.

A. Uskov and H. Avagyan [23] analysed the performance of two phase authenticated encryption associated data (AEAD) cryptographic algorithm Galois/Counter Mode (GCM) and compared its performance with AES, RC6, TwoFish and Camellia cipher algorithms. The analysis results revealed that AES-GCM outperforms the other combinations, but they analysed only GCM algorithm. A. Tanveer *et al.* [24] analysed the performance of IPSec using combination of AES-finalists proposed algorithms (Rijndael, Serpent, Twofish, RC6 and Mars) encryption algorithms and secure hashing algorithms (MD5, SHA1 and SHA2). They modified Linux kernel and IPSec software module to support missing cryptographic algorithms, but they analysed only AES-finalist variants in IPv4 network. L. Lian and G. Wen-mei [25] explained the method to implement IPSec in IPv6 based network environment using Openswan, but they did not do any performance analysis on IPSec protocol and associated cryptographic algorithms.

3 IPSec Standards and Recommendations

3.1 IETF IPSec Recommendations

Internet Engineering Task Force (IETF) officially standardizes IPSec development in a series of RFCs. IPSec v3 series (starts with RFC4301) of RFC documents are the latest updated versions explain the IPSec implementation methods from standards perspective. The roadmap for IPSec and IKE protocols are explained in [8]. IPSec standard is developed in such a way that cryptographic algorithms can be chosen independently by users and system administrators irrespective of the protocols used to implement

IPSec. The recommendations for using cryptographic algorithms in IPSec architecture is updated in timely to ensure the secureness of IPSec process and implementation. The research study and also standards recommend using ESP protocol rather than AH protocolling. ESP protocol in tunnel mode provides equivalent security features of tunnel mode AH protocol. The latest recommendations for using cryptographic algorithms in IPSec architecture is defined in [13]. In [26], cryptographic algorithms for use in IKE v2 details are explained.

3.2 3GPP IPSec Recommendations

Since the 3rd generation (3G) of mobile networks in 3GPP, network domain security (NDS) feature is included in the telecommunication network, where network security aims to secure the communication between Network Elements (NEs). In 2G, General Radio Service (GPRS), and 3G, Universal Mobile Telecommunications System (UMTS), only data communication in the Packet Switched (PS) domain is based on IP, but also offer non-IP services in the Circuit Switched (CS) domain. The 4th generation (4G), the Evolved Packet System (EPS), is totally IP based without any CS domain, while moving traditional CS services in to the IP Multimedia Subsystem (IMS). IPSec is preferred to secure the control signalling on selected interfaces between 3GPP network elements using NDS/IP specifications, including signalling of the IMS at the application layer.

The architecture of Network Domain Security for IP (NDS/IP) [14] is shown in Figure 1. Here two network security domains are considered that

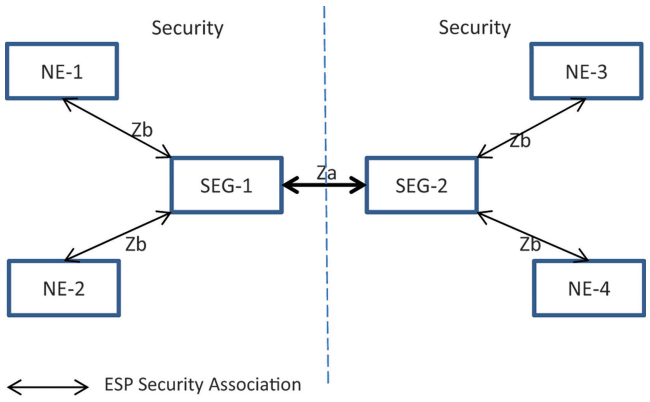

Figure 1 Network domain security for IP based protocols [14].

may belong to the same operator or different operators. The network security domains are networks that are managed by a single administrative authority and consist of various network elements. IPSec can be implemented in the Za and Zb interfaces. Use of ESP and a set of cryptographic algorithms are recommended. The selection of cryptographic algorithms for IPSec implementation is operator's choice. Security Gateways (SEGs) directly communicating with other network security domain entities in Za interface; so SEGs will maintain at least one IPSec tunnel in tunnel mode between them. NEs can also establish and maintain IPSec secured ESP SA within the network security domain.

In 3GPP networks, GPRS Tunnelling Protocol (GTP) is used to carry both control plane data (GTP-C) and user plane data (GTP-U) from core network to access network and vice versa. Control plane data are comparatively more sensitive, so IPSec is preferred to secure control plane data (GTP-C). IPSec ESP protocol in tunnel mode is recommended. On the Za interface (SEG-SEG) authentication/integrity protection is mandatory and confidentiality is recommended. On the Zb interface (SEG-NE/NE-NE) authentication/integrity protection is mandatory and confidentiality is optional. In this work, we have considered the confidentiality and integrity algorithms recommended by standards for the performance study of IPSec. 3GPP TS 33.210 also refer the IETF RFC 7321 for using cryptographic algorithms in IPSec framework.

4 Experimental Setup

For providing secure communication between two network entities secure tunnel can be created using different protocols (IPSec, Secure Socket Layer/ Transport Layer Security (SSL/TLS), PPTP, L2TP). IPSec based tunnels provide better security and are quite common in the network world. IPSec can be implemented by using either software or hardware resources. Compared to software implementation, hardware implementations are better because they can accelerate the performance level and also match the lengthy identities in Security Association Database (SAD). Routers or firewalls are used as gateways in the hardware based IPSec implementation. Several vendors provide hardware VPN, such as Cisco, IBM, Juniper, Checkpoint, etc. For software IPSec implementation, open source software tools are available such as Openswan, Strongswan, Libreswan, etc.

In this experimental work, we use Strongswan [27] software tool (version 5.3.5) to implement IPSec. Charon is a Strongswan IKE daemon which supports both IKEv1 and IKEv2. There are many plugins to support various features such as openssl, gcm, ccm, xcbc, sha3 algorithms, etc. The Charon

daemon has access to configuration files, keys, certificates and other files, if required. It uses IKE protocol to establish SAs and to negotiate the cryptographic algorithms to be utilized by the IPSec stack which resides in the kernel.

Here Internet Key Exchange version 2 (IKEv2) protocol is employed for cryptographic algorithms negotiation and key exchange. Strongswan supports various authentication techniques such as pre-shared keying, X.509 certificate based authentication, EAP based certificate less authentication, etc. For the experiments, we use Elliptic Curve Digital Signature Algorithm (ECP256) for X.509 certification based authentication between communicating entities. Two sets of cryptographic algorithms are defined in IPSec configuration file. One set of algorithm functions to protect the IKE protocol communication and its parameters and another set functions to protect the actual data communicated over the medium between the authenticated entities. Further we have employed the combination of AES128-SHA256-ECP256 cryptographic algorithms for use in the IKEv2 [26].

4.1 Hardware Setup

IPSec experimental setup is shown in Figure 2. Two Linux OS based physical systems were used to create IPSec tunnels between them by using Strongswan version 5.3.5, both the systems host Ubuntu 14.04 LTS operating system and their specifications are as given in Table 1. We installed Strongswan in both of these systems to setup the IPSec tunnel. In this experiment a 1Gbps Network Interface Card has been used between the two systems to study the performance of IPSec in a host to host (gateway to gateway) network.

4.2 Measurement Tools

In this work we have used iperf3 an open source tool to analyse the performance of IPSec. Iperf3 was used for generating the test traffic for testing

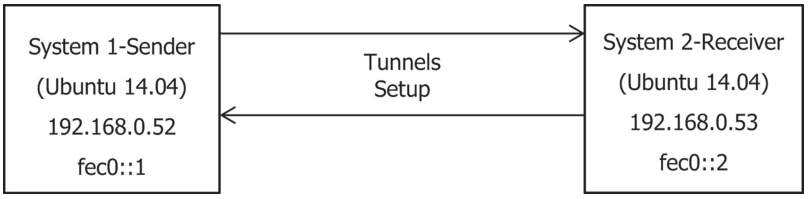

Figure 2 IPsec experimental setup.

Table 1 Setup systems specification details

Specs	System 1	System 2
CPU	8 cores	4 cores
Memory	8 GB RAM	8 GB RAM
HDD	500 GB	1 TB

the IPSec tunnel throughput under various encryption and authentication algorithms.

4.3 Testing Parameters

The following are the input parameters used in our performance study.

ESP combined mode algorithms provides both confidentiality and authentication services; in cryptographic terms, these are AEAD algorithms [RFC5116]. They are given in Table 2. Tables 3 and 4 list out the Encryption and Authentication Algorithms that are recommended in the RFC 7321, which is also recommended by 3GPP [14].

The reason to analyze the performance of IPSec in various scenarios is, because as mentioned in the 3GPP requirement, based on circumstances various implementations are possible, for example in Zb interface (shown in Figure 1) authentication is mandatory but confidentiality is optional as per the

Table 2 ESP Authenticated Encryption (Combined Mode Algorithms)

Requirement	Authenticated Encryption Algorithm
SHOULD+	AES-GCM with a 16 octet ICV [RFC4106]
MAY	AES-CCM [RFC4309]

Table 3 ESP Encryption Algorithms

Requirement	Encryption Algorithm
MUST	NULL [RFC2410]
MUST	AES-CBC [RFC3602]
MAY	AES-CTR [RFC3686]
MAY	TripleDES-CBC [RFC2451]
MUST NOT	DES-CBC [RFC2405]

Table 4 ESP Authentication Algorithms

Requirement	Encryption Algorithm
MUST	HMAC-SHA1-96 [RFC2404]
SHOULD+	AES-GMAC with AES-128 [RFC4543]
SHOULD	AES-XCBC-MAC-96 [RFC3566]
MAY	NULL [RFC4303]

3GPP specifications. For this purpose we have studied the scenarios for all variations so as to get a complete picture of performance impact.

We have used TCP and UDP for testing the performance over IPSec tunnels in both IPv4 and IPv6 network environment.

4.4 Other Input Parameters

For the purpose of this experiment, we have tested the IPSec performance only for ESP protocol with tunnel mode setup. One of our main goals was to measure the performance of the newly proposed AEAD algorithms (AES128-GCM16, AES128-CCM16), encryption and authentication algorithms for both IPv4 and IPv6.

We measured the throughput using iperf3; we took at least three readings for all combinations and reported the mean value. Test data was transmitted from client to server for 60s.

5 Results and Discussions

Table 5 compares the IPSec performance in both IPv4 and IPv6. We observed that UDP performs better than TCP. TCP has a header size of 20 bytes as compared to the 8 bytes in UDP which can be attributed to better performance. Also UDP being a connectionless protocol doesn't have any acknowledgement messages thus involving lesser overhead as compared to TCP.

Figure 3 shows the performance of various IPSec combined mode cryptographic algorithms in IPv4 network. We observed that AES128-GMC16 performance is the best among the AEAD algorithms. The performance

Table 5 Performance in IPv4 and IPv6 network

	IPv4		IPv6	
Encryption/ Authentication	Throughput (TCP)	Throughput (UDP)	Throughput (TCP)	Throughput (UDP)
No IPSec	935	951	921	933
AES128-GCM16	899	942	873	885
AES128-CCM16	773	817	737	807
AES128-SHA1_160	839	931	804	826
AES128-SHA2_256	710	835	637	643
AES128CBC-NULL	903	945	878	888

Note: All Throughput measurements are in Mbps.

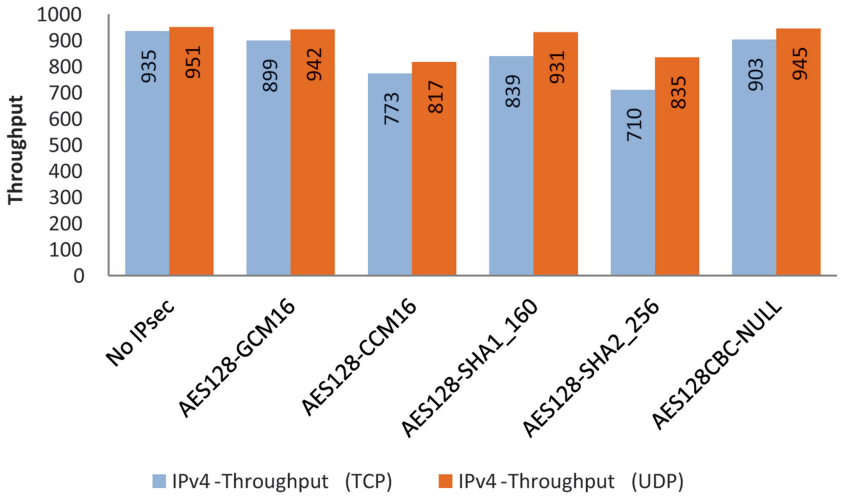

Figure 3 Encryption and authentication algorithms performance in IPv4.

Note: All Throughput measurements are in Mbps.

of AES128-GCM16 is very slightly lower when compared to AES128-NULL. But AES128-GCM16 provides both confidentiality and authentication whereas AES128-NULL is an encryption only algorithm.

Another observation is that SHA1 performs better as compared to SHA2 for the same encryption algorithm. But since SHA1 is already proved to be a weak algorithm [11] it is suggested that it should not be used.

Figure 4 shows the performance comparison of combined algorithms in both IPv4 and IPv6 network environment. From the results we noticed that IPv4 performs slightly better when compared to IPv6. We observed that for all AEAD algorithms that were tested IPv4 performance is better this might be attributed to the extra overhead in the IPv6 header as compared to IPv4. IPv4 header has a variable header length from 20 bytes to 60 bytes, whereas the IPv6 header is a constant 40 bytes.

Table 6 compares the UDP packets performance for different cryptographic algorithms. From Table 6, we observe that in case of UDP the performance of the AEAD algorithms and combination of other encryption and authentication algorithms is quite similar to TCP, where the throughput is slightly better for IPv4 as compared to IPv6. UDP being a connectionless algorithm has higher chances of errors occurring while transmission. We observed that changing the algorithm gives slight differences in the jitter. We

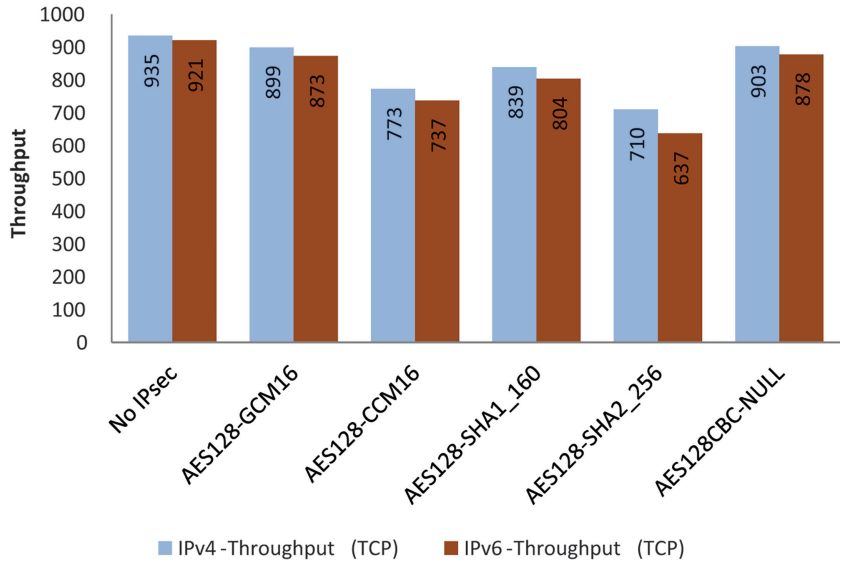

Figure 4 Combined mode algorithms performance in IPv4 and IPv6.

Note: All Throughput measurements are in Mbps.

Table 6 Comparison of various algorithms for UDP

	IPv4			IPv6		
Encryption/ Authentication	Throughput (UDP)	Jitter (ms)	Packet Loss (%)	Throughput (UDP)	Jitter (ms)	Packet Loss (%)
No IPSec	951	0.065	0.00023	933	0.091	0
AES128-GCM16	942	0.06	0	885	0.077	0
AES128-CCM16	817	0.105	1.4	807	0.107	1.6
AES128-SHA1_160	931	0.073	0.86	826	0.076	0.54
AES128-SHA2_256	835	0.121	2.4	643	0.123	0.6
AES128CBC-NULL	945	0.073	0.0016	888	0.076	0.00098

Note: All Throughput measurements are in Mbps.

also observed that AES-128-CCM16 showed similar packet loss for IPv4 and IPv6. We would also like to mention that these performance parameters like jitter and loss depend on the network connection and congestion parameters, so it may not be same all the time, we have tried to take the average of the readings. But these may significantly vary depending on the experimental setup and other real time factors.

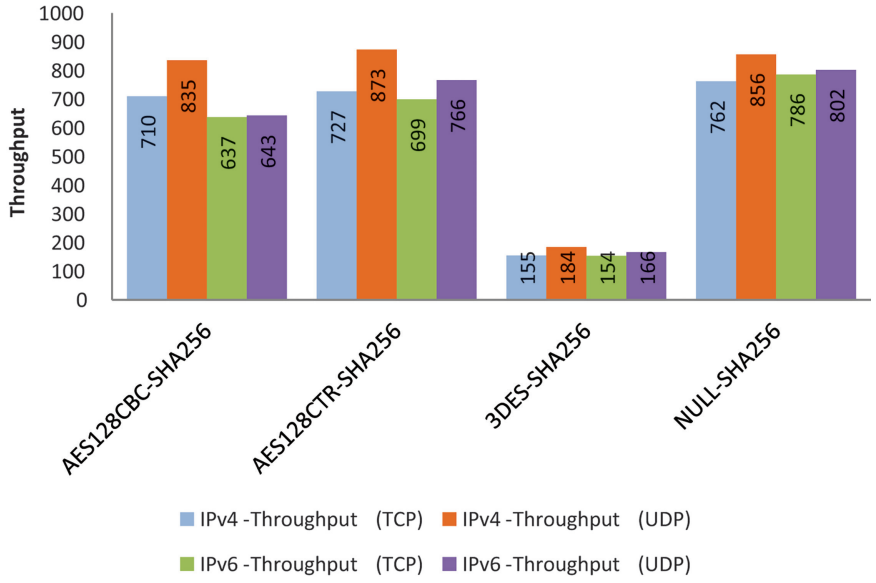

Figure 5 Performance for varying encryption algorithms.

Note: All Throughput measurements are in Mbps.

Figure 5 depicts the performance comparison while changing only the encryption algorithm for a constant authentication algorithm i.e. SHA2. From the graph it can be inferred that AES performs better than 3DES this can be attributed to the fact that AES performs a single encryption operation while 3DES repeats the DES block encryption three times which severely impacts its performance. AES128-CTR (Counter Mode) performs better as compared to AES128-CBC (Cipher Block Chaining) in both TCP and UDP mode of operation. AES128CTR also performs better for IPv4 in comparison with IPv6.

The jitter observed for UDP in IPv4 and IPv6 are somewhat similar as observed in Table 7. The packet loss observed was higher for IPv4 than in IPv6. Similar results observed for AEAD algorithms where the throughput for IPv4 is better as compared to IPv6. And 3DES-SHA256 showed poor performance in UDP as well as compared to the other encryption algorithms.

Figure 6 shows the performance for various authentication algorithms applied. It is seen that SHA1 performs slightly better than SHA2 for the same encryption algorithm. There is another authentication algorithm used called AESXCBC, it uses AES in CBC mode with a set of extensions for MAC

Table 7 UDP performance for Varying Encryption Algorithms

	IPv4			IPv6		
Encryption/ Authentication	Throughput (UDP)	Jitter (ms)	Packet Loss (%)	Throughput (UDP)	Jitter (ms)	Packet Loss (%)
AES128CBC-SHA256	835	0.121	2.4	643	0.123	0.6
AES128CTR-SHA256	873	0.108	1.8	766	0.115	3
3DES-SHA256	184	0.5	5.3	166	0.336	2.6
NULL-SHA256	856	0.107	1.4	802	0.137	0.24

Note: All Throughput measurements are in Mbps.

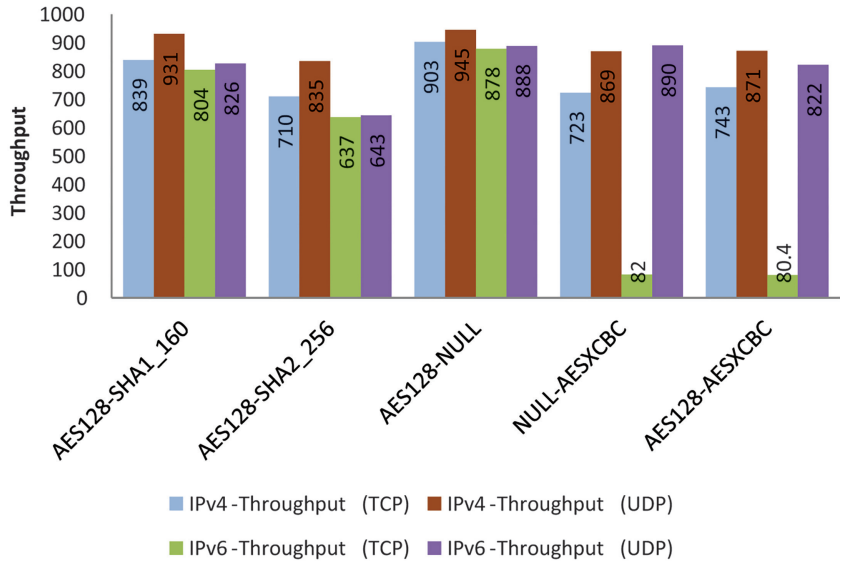

Figure 6 Performance for varying authentication algorithms.

Note: All Throughput measurements are in Mbps.

based on way hash function which is helpful in securing messages of varying lengths. We observed that AES128-AESXCBC performed better than NULL-AESXCBC for IPv4. Comparatively in IPv6 although the results were quite similar but on an average NULL-AESXCBC performed better.

An anomalous behaviour in case of AESXCBC algorithms when running IPv6 in TCP mode was observed. We saw that the throughput in this scenario was drastically reduced of the magnitude of 10x as compared to IPv4. The phenomena persisted even after changing the encryption algorithms (i.e. AES128, NULL).

Table 8 UDP performance for different cryptographic algorithms

	IPv4			IPv6		
Encryption/ Authentication	Throughput (UDP)	Jitter (ms)	Packet Loss (%)	Throughput (UDP)	Jitter (ms)	Packet Loss (%)
AES128-SHA1_160	931	0.073	0.86	826	0.076	0.54
AES128-SHA2_256	835	0.121	2.4	643	0.123	0.6
AES128CBC-NULL	945	0.073	0.0016	888	0.076	0.00098
NULL-AESXCBC	869	11.025	5.9	890	0.149	30
AES128-AESXCBC	871	10.951	3.2	822	0.116	35

Note: All Throughput measurements are in Mbps

The UDP performance observed for varying the cryptographic algorithms was tabled in Table 8. SHA1_160 performing better than SHA256. The performance of AESXCBC is also seen to be quite good in comparison to other authentication algorithms but the results showed that the packet loss in case of AESXCBC was quite large as compared to the others. Packet loss observed in case of AESXCBC for IPv4 was around 5% while for IPv6 it was 30% or more which was higher as compared to their other counterparts. Our observation also showed that when using AESXCBC it causes more packet loss and jitter in the network, the reason for this anomalous behaviour is under study. But based on our results we can say that of the studied algorithms AES128-AESXCBC performs the best in terms of performance and security as compared to the other authentication algorithms.

We noticed that the throughput varies based on the encryption algorithm; this can be attributed to the complexity and processing delay incurred by the system in encrypting and transmitting the complete data.

Our intention of using a direct Ethernet cable between the systems was to ensure that maximum data can be transmitted over the network without too much interference and loss, but we observed some minor packet losses and jitter throughout our observation. It is our view that this can be attributed to the properties of the cable used to connect the systems.

6 Conclusion

We have analysed the throughput performance of IPSec using various combined mode cryptographic algorithms, encryption algorithms and authentication algorithms as suggested by standards recently for implementing IPSec in NDS/IP networks. The Open source tool Strongswan was used in this work for the IPSec implementation and we presented the corresponding

configurations and setup details. The performance of IPSec was analysed in three different cases based on the algorithms that were chosen. The analysis results concluded that in AEAD algorithms AES128GCM16 performs better than AES128CCM16, AES-CTR performs better than AES-CBC and 3DES; and SHA1 performs better than SHA256. We plan to investigate extensively the performance of AESXCBC algorithms and the reason for its anomalous behaviour in particular scenarios as a future work.

References

[1] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301, Dec. 2005.

[2] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, Nov. 1998.

[3] C. Kaufman *et al*., "Internet Key Exchange Protocol Version 2 (IKEv2)," RFC 7296, Oct. 2014.

[4] D. Maughan *et al*., "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, Nov. 1998.

[5] H. Orman, "The OAKLEY Key Determination Protocol," RFC 2412, Nov. 1998.

[6] S. Kent, "IP Authentication Header," RFC 4302, Dec. 2005.

[7] S. Kent, "IP Encapsulating Security Payload," RFC 4303, Dec. 2005.

[8] S. Frankel and S. Krishnan, "IP Security (IPSec) and Internet Key Exchange (IKE) Document Roadmap," RFC 6071, Feb. 2011.

[9] S. Deering and R. Hinden, "Internet Protocol Version 6 (IPv6)," RFC 2460, Dec. 1998.

[10] S. Deering and R. Hinden, "IP Version 6 Addressing Architecture," RFC 4291, Feb. 2006.

[11] M. Stevens *et al*., "Freestart collision for full SHA-1," *Cryptology ePrint Archive, Report 2015/967*, 2015.

[12] XiaoyunWang and Hongbo Yu, How to Break MD5 and Other Hash Functions, *EUROCRYPT (Ronald Cramer, ed.), Lecture Notes in Computer Science, vol. 3494, Springer,* 2005, pp. 19–35.

[13] D. Mcgrew and P. Hoffman, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)," RFC 7321 (Obsoletes RFC 4835), Aug. 2014.

[14] 3GPP TS 33.210: "3G security; Network Domain Security; IP network layer security," 2015.

[15] N. Ferguson and B. Schneier, "A Cryptographic Evaluation of IPSec", 2003. Available: http://www.schneier.com/paper-IPSec.pdf

[16] K.G. Paterson and A.K.L. Yau, "Cryptography in Theory and Practice: The Case of Encryption in IPSec." *In S. Vaudenay (ed.), EUROCRYPT 2006, LNCS Vol. 4004, Springer,* 2006, pp. 12–29.

[17] J. L. Degabriele and K. G. Paterson, "On the (In) Security of IPSec in MAC-then-Encrypt Configurations," In *Proc. of the 17th ACM conference on Computer and communications security*, 2010, Pages 493–504.

[18] O. Elkeelany *et al*., "Performance Analysis of IPSec: Encryption and Authentication," In *Proc*. *IEEE Inter. Conf. on Communications*, 2002, pp. 1164–1168.

[19] G. Hadjichristofi *et al*., "IPSec overhead in wireline and wireless networks for Web and email applications," In *Proc. IEEE Inter. Conf. on Performance, Computing, and Communications*, 2003, pp. 543–547.

[20] C. Xenakis, *et al*., "A generic characterization of the overheads imposed by IPSec and associated cryptographic algorithms," *Computer Networks, Volume 50, Issue 17*, 2006, pp. 3225–3241.

[21] C. Shue *et al*., "Analysis of IPSec overheads for VPN servers," IEEE ICNP's NPSec Workshop, 2005.

[22] C. Shue *et al.*, "IPSec: Performance Analysis and Enhancements," In *Proc. IEEE Inter. Conf. on Communications*, 2007, 1527–1532.

[23] A. Uskov and H. Avagyan, "The Efficiency of Block Ciphers in Galois/Counter Mode in IPSec-Based Virtual Private Networks," In *Proc. IEEE Inter. Conf. on Electro/Information Technology*, 2014, 173–178.

[24] A. Tanveer *et al*., "Performance Analysis of AES-Finalists along with SHS in IPSec VPN over 1Gpbs Link," In *Proc*. *IEEE Inter. Bhurban conf. on Applied Sciences and Technology*, 2015, pp. 323–332.

[25] L. Lian, and G. Wen-mei, "Building IPSec VPN in IPv6 Based on Openswan," In *Proc. IEEE Inter. Conf. on Network and Parallel Computing Workshops*, 2014, 173–178.

[26] J. Schiller, "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)," RFC 4307, Dec. 2005.

[27] Strongswan IPSec software tool. Available: https://strongswan.org/

Biographies

Prabhu Thiruvasagam received master degree in Communication Systems from Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram, India in 2014. Then, he worked two plus years as research engineer in NEC India Standardization Team at NEC India Pvt Ltd, Chennai. Now, he is pursuing PhD in the department of Computer Science at Indian Institute of Technology Madras, India. Currently, his research interest includes Security and Reliability aspects of NFV, SDN, and SFC paradigms in 5G networks.

K. Jijo George received his Bachelors in Computer Science and Engineering from Kurukshetra Institute of Technology and Management, India in 2011. He has over 4 years of experience in research and development of mobile communication networks. He worked as Research Engineer in NEC India Standardization (NIS) Team at NEC Technologies India Private Ltd. Chennai. Prior to joining NECI he was associated with IIIT, Bangalore as Research Associate in Context awareness in mobile applications. At NEC he worked

on security aspects of telecom networks and testbed development of next generation mobile networks. His research interest includes Next Generation Networks, Mobile and Network Security and Telecom Security. He is currently pursuing his Masters in Cognitive Technical Systems in Albert Ludwigs University Freiburg, Germany.