Available online at www.ejournals.uofk.edu

UofKEJ Vol.5 Issue 1 pp. 10-17 (Febrauary-2015)

UNIVERSITY OF KHARTOUM ENGINEERING JOURNAL (UOFKEJ)

VSAT Packet Security Performance Enhancement Using Modified Interlock Protocol

Ayman M. Muzzmail, Hamid Abbas Ali, Sami M. Sharif

[1]*Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Khartoum Khartoum, Sudan (E-mail: aymanmuz1@hotmail.com)*

Abstract: This study has focused on developing the Interlock protocol and on implementing it on the VSAT system to overcome the man-in-the-middle attack in the double hob topology. A new cryptography protocol has been modified into Interlock protocol by using the concept of Cipher Block Chinning mode and AES, RSA, SAH-256 algorithms. The TCP protocol is flexible and has the ability to tune its parameters depending on the environment under which it works. In this study the TCP protocol parameters had been readjusted to reach the optimum throughput when the packet was encrypted and transmitted in the VSAT environment. The results indicate an increase in the throughput of up to about 30%.

Keywords*: VSAT system; Man-in-the-middle attack; Interlock protocol; Throughput*.

1. INTRODUCTION

The packet security in satellite communications still did not meet the requirements because of the high delay. A propagation delay is incurred when packets cross the estimated distance of about 36000 kilometer for one hop from source to destination in GEO [1]. The inherent delay is around 540 ms and may reach 800 ms in one hop. Its computation becomes more complicated when using double hop topology in the VSAT system.

In VSAT topology the main part of satellite hub is used for management purposes and all data will move through it. This would be appropriate for the man-in-the-middle to attack the data. Attacks would be more effective if they are active. In the scenario created here, the attackers will work on making the sender and receiver believe that they talk to each other. In the VSAT environment, the traditional cryptography methods that are based on using trusted third party concept or CA are unsuitable due to their impact on the VSAT network performance.

CA needs to be placed as a terminal station in the VSAT network system and whenever other stations require communicating with each other they must ask for the public key [2]. This situation complicates the propagation delay problem because of the great distance of 36000 km that is actually traveled by the signal between the stations and the satellite in GEOs. To understand the active attack by the man-in-the-middle, assume that there are two terminals in a VSAT system represented by Alice and Bob. Mallory is at the main point or hub station in a VSAT company provider. The attack works as follow:
Alice sends Bob her public key. Mallory intercepts this key and sends Bob his own public key. Bob sends Alice his public key. Mallory intercepts this key and sends Alice his own public key. When Alice sends a message to Bob, encrypted in "Bob's" public key, Mallory intercepts it. Since the message is really encrypted with his public key, he decrypts it with his private key, re-encrypts it with Bob's public key, and passes it on to Bob. When Bob sends a message to Alice, encrypted in "Alice's" public key, Mallory intercepts it. Since the message is encrypted with his public key, he decrypts it with his private key, re-encrypts it with Alice's public key, and forwards it on to Alice [3].

2. INTERLOCK PROTOCOL

The interlock protocol, as described by Ron Rivest and Adi Shamir, was designed to frustrate eavesdropper's attacks against two parties that use an anonymous key exchange protocol to secure their conversations [4]. It works as follow:
Alice sends Bob her public key. Bob responds by sending Alice his public key. Then Alice encrypts her message using Bob's public key and sends half of the encrypted message to Bob. Bob encrypts his message using Alice's public key and sends half of the encrypted message to Alice. Alice will send the other half of her encrypted message to Bob. Bob puts the two halves of Alice's message together and decrypts it with his private key. Bob sends the other half of his encrypted message to Alice. Alice will put the two halves of Bob's message together and decrypts it with her private key.

This protocol maybe useful for authenticating passwords between users and a host as proposed and explained by Davies and Price [5]. However, it seems very weak in case of exchanging keys for encrypted data without modifying it.

3. DEVELOPMENT OF THE MODIFIED INTERLOCK PROTOCOL

To realize this Interlock Protocol, it is assumed that a Cipher Block Chaining mode is used. This proposed Interlock Protocol will be referred to as the modified Interlock Protocol or MIP. The protocol is based on dividing the message into three parts in the encryption phase as shown in **Fig. 1.** The first part is the original message; it is XORed with the initialization vector and is decrypted later on by the AES algorithm. The outcome of this part is called Cipher 1. In the second part, the message is hashed using the SHA algorithm with 256 bits. After that it's XORed with Cipher1 and encrypted with AES algorithm to provide the second part which is called Cipher2. Finally, the third part involves the encryption of the AES's key by using RSA algorithm. This part is called Key encrypted Key or KEK.

The decryption phase is the reverse of the Cipher Block Chaining mode. In this phase, the receiver can use the hash message to know if there was a change in the message by the man-in-the-middle. **Fig. 2** illustrates the decryption phase.

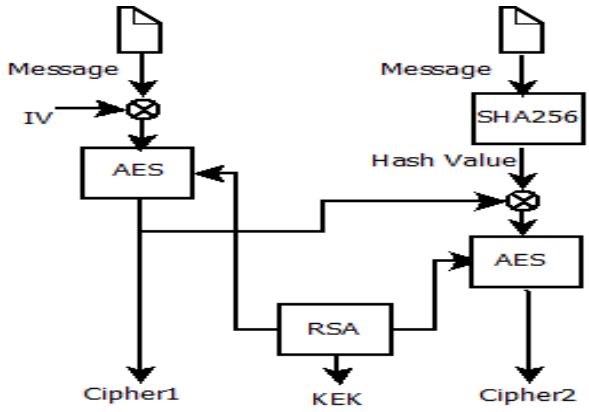


Fig. 1. The modified Interlock Protocol (MIP) in the encryption phase

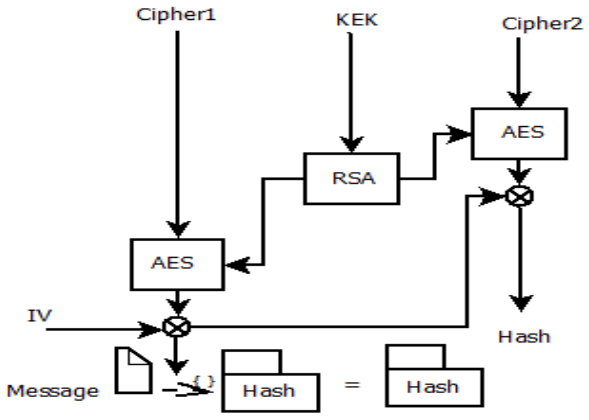


Fig. 2. The Modified Interlock Protocol (MIP) in the decryption phase

This situation makes the man-in-the-middle facing a real problem when he (she) intercepts one part of the message; such as Cipher1, Cipher2 or the KEK. This is because the intercepted part cannot be decrypted with the deceptive private key, neither encrypted, as it should be, with the public key. In this case, the interceptor has no other choice but to send it, as it is, before receiving the other parts of the message.

The MIP protocol has been coded in Java platform. The **javax.crypto** package specially designed for cryptography is used here. In order to create a Cipher object, the application calls the Cipher's **getInstance()** method, and passes the name of the requested transformation to it. The flow chart for implementing the MIP is shown in **Fig. 3** for the encryption phase and in **Fig. 4** for the decryption phase.

4. MIP PERFORMANCE IN THE VSAT NETWORK

There are two kinds of delay that has a direct impact on the VSAT network performance implementing MIP; propagation delay and transmission delay [6]. The propagation delay is the amount of time it takes for the head of the signal to travel from the sender to the receiver. It is given by:

$$\text{Propagation delay} = d / c \quad (1)$$

where d is the distance and c represents speed of light.

The VSAT network system normally creates long delays due to the distance that is traveled by the signal in two hops (the satellite orbit is 35,786 kilometers from the earth). The MIP will increase the transmission delay because it adds more bits for encryption in the packet payload.

Transmission delay is caused by the data-rate of the link and is calculated by

$$D_T = N/R \quad (2)$$

where: D_T is the transmission delay in seconds,
N is the number of bits and R is the rate of transmission in bits per second.

MIP is working under TCP protocol for transmitting segments. TCP is flexible protocol and it has ability to work in different environmental networks. It has a lot of mechanisms and parameters that can be readjusted to achieve an acceptable throughput. This study has focused on Received Window Size, Slow Start Initial Count, Fast Retransmission, Fast Recovery, Initial Retransmission Timeout (RTO), Window Scaling Option and Selective ACK (SACK) to achieve the optimum throughput for the VSAT network.

The flow control is a mechanism in TCP implemented in the receive side. It instructs the sender about how much data it is willing to accept. TCP employs a sliding window algorithm for flow control. On the sender side it has a buffer which is used to keep the data that has been sent but not acknowledged

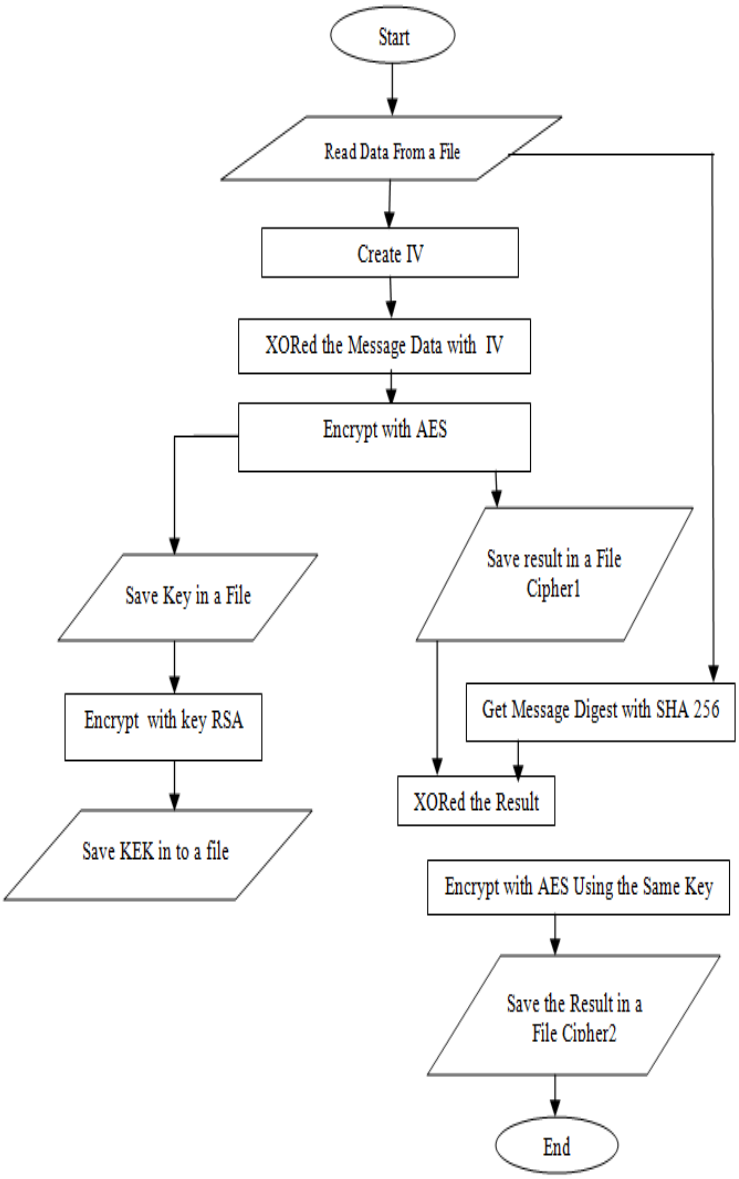


Fig. 3: The MIP Model in the encryption phase

yet. On the receive side, TCP maintains a receive buffer which holds the received data that hasn't yet been read. To avoid being running out of buffer space, the receiver uses a window size to clearly declare how much the data size that can be accepted, which depends on its buffer size [7].

RFC-2414 [8] had recommended that for high performance of throughput, maximum segment size is used as a reference to initialize slow-start phase as follow:
If (*MSS* <= 1095 bytes)

Then (initial *cwnd* = 4*MSS)

If (1095 bytes < *MSS* < 2190 bytes)

Then (initial *cwnd* = 4380)

If (2190 bytes <= *MSS*)

Then (initial *cwnd* = 2*MSS)

The Maximum Segment Size (MSS) is a parameter specifies the largest amount of data, in octets, that a computer or communication device can receive in a single TCP segment [9].

TCP uses retransmission timer which let the sender to stop and wait for a specified time before it resends the packet that gets lost. This specified time is called retransmission timeout or (RTO). RFC 6298 proposed the calculation of retransmission timeout [10]. It's suggested that the TCP sender maintains two state variables SRTT (smoothed round-trip time) and RTTVAR (round-trip time variation). After that it combines a smoothed estimate of the RTT with the main deviation of the RTT to obtain the RTO estimate. Initial retransmission timeout value is a first estimate for the RTO before it is updated and came to effect. The Initial RTO is a sensitive parameter. Intensive study had suggested the choice of 3 seconds.

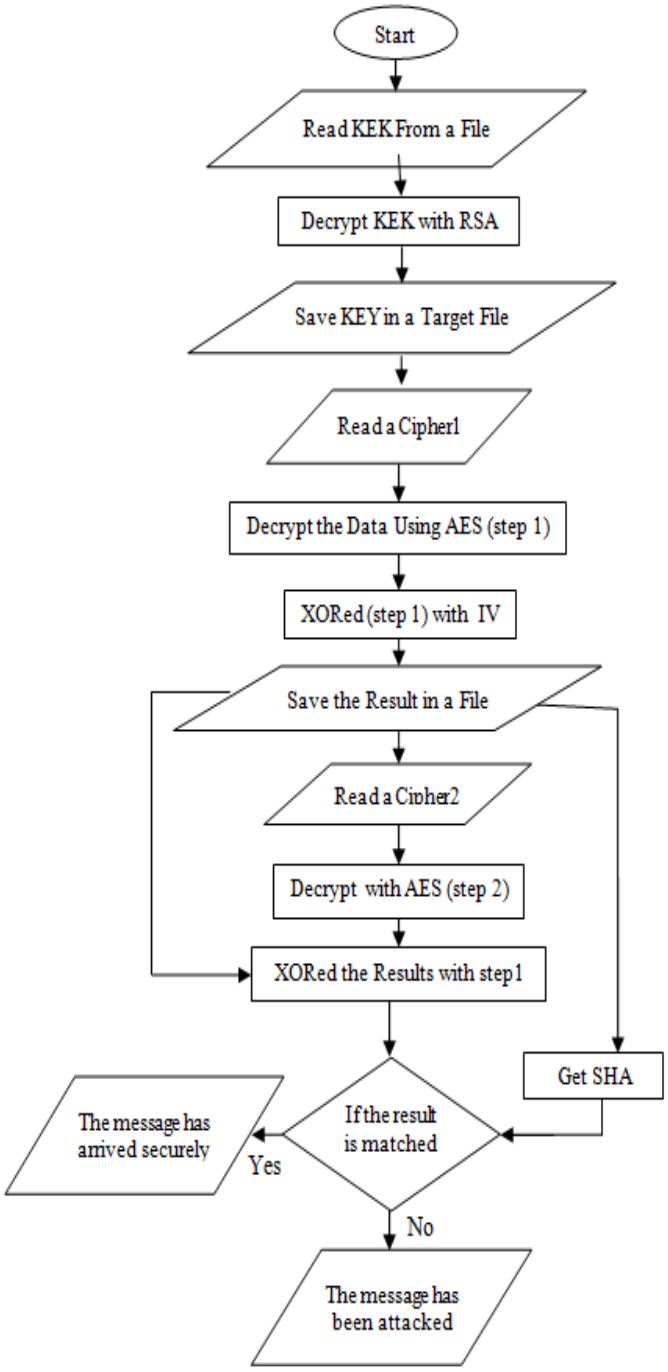


Fig. 4. The MIP Model in the decryption phase

In fast retransmission the sender resends the packet that is lost without waiting for the timer when it has received three duplicate acknowledgments from the receiver side. TCP Tahoe is a version of the fast retransmission algorithm. When the fast retransmit mechanism has been used and assuming that there is congestion in the network the sender starts from the congestion avoidance phase rather than going into the slow-start phase which is usually the case. This kind of mechanism is called fast recovery. TCP New Reno is a version that implements both fast retransmit and fast recovery together [11].

Selective Acknowledgement (SACK) is an optional field in the segment which solves the problem of delay in fast retransmits and fast recovery algorithms. SACK allows the receiver to inform the sender exactly which of the segments have arrived. This kind of mechanism reduces the spent time by using other mechanisms [12].

The TCP window scale option is the ability to increase the receive window size allowed in TCP above its former maximum value of 65,535 bytes. In data communications, bandwidth-delay product or (BDP) refers to the product of a data link's capacity (in bits per second) and its round-trip

delay time (in seconds). The result is an amount of data measured in bits (or bytes), that is equivalent to the maximum amount of data on the network circuit at any given time, i.e., data that has been transmitted but not yet acknowledged. The TCP window scale option is needed for efficient transfer of data when the bandwidth-delay product is greater than 64K.

5. ENVIRONMENTAL LAB

The MIP model that has just been discussed was implemented in an environmental lab**.** The VSAT system has been installed with three stations. Two terminals and one hub or min point. The system of VSAT is working in Ku band. Outdoor units consist of Antenna with 2.4 meter dimensions, Block UpConverter (BUC) with 4 watt and Low Noise Block Downconverter (LNB) with local oscillator frequency of 10GHz. In door unit consists of Comtech CDM-576 L-Band Satellite Modem. At the terminal in station A an FTP Server with FTP Daemon software has been installed and SD TCP Optimizer software. In station B Wireshark capture software has been installed with SD TCP Optimizer software. The Hub station consists of additional module that is a Comtech CDD-564L L-Band satellite demodulator indoor and external power unit of 48 Watt outdoor. **Fig. 5** shows the structure of the VSAT system.

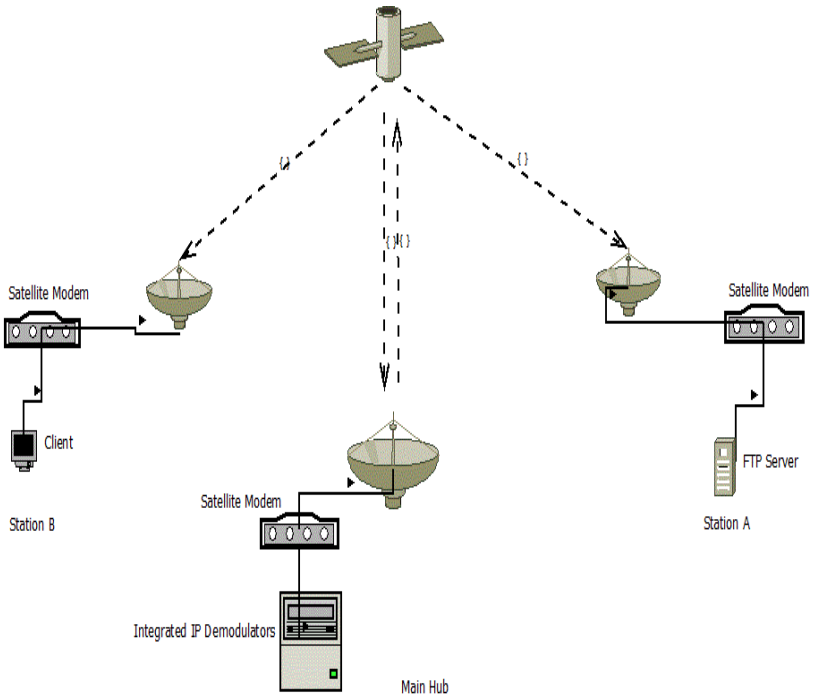


Fig. 5. VSAT system Lab environment

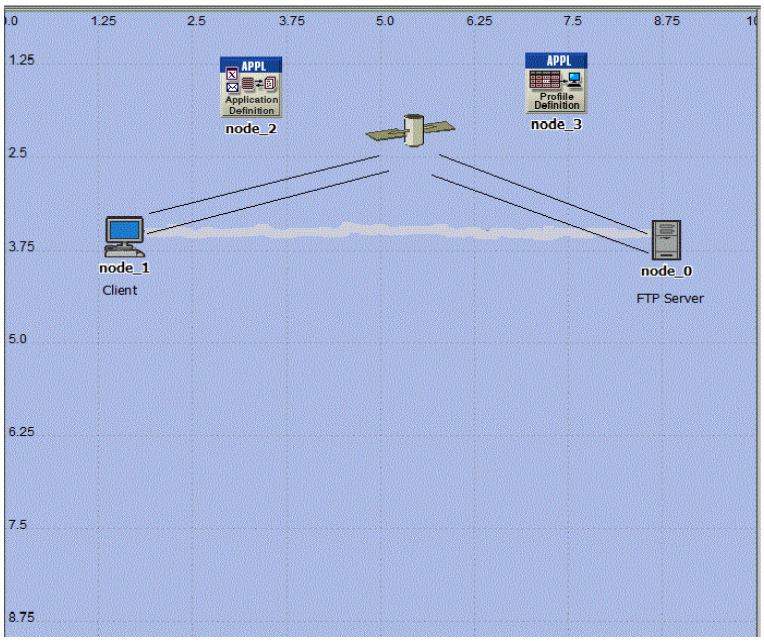


Fig. 6. The Simulation Environment

In this experimental setup, a file of size 100 KB has been encrypted by using MIP and three files were produced; i.e. Cipher1 with 194 KB, Cipher2 with 160 Byte and KEK with 259 Byte. These three files had been sent over VSAT satellite system by using a File Transfer Protocol (FTP) between station A and station B with average round trip time RTT of 591ms. Those three files were sent ten times. The average times taken were; Cipher1 was sent in 53.93 seconds, Cipher2 was sent in 6.47 seconds and KEK was sent in 6.45 seconds. The time between sending and receiving was calculated using Wireshark software. The behavior of the environment was studied using latency calculator and MTU calculator tools in SG TCP Optimizer software. The MTU was 1500 while the MSS was 1471 as derived from it. The average throughput was 24101 bit/s.

6. OPNET SIMULATOR

OPNET simulator had been used, depending on the real environmental lab, to tune the TCP protocol already discussed. In this simulator the results at the environmental lab in section (5) had been considered as the main guidance. In this study, OPNET Modeler 14.5 is configured as follows:

1- Two stations (FTP server and Client) had been established in the simulator as shown in **Fig. 6**.
2- A point to point advance link is used to connect the two stations. After that the speed of data transmission over the link was specified as 64 Kbit/s, choosing the propagation speed of light mode as illustrated in **Fig. 7.**
3- The behavior of the link for the data transmission rate as in the real environmental lab, section (5), was setup and exported to the simulator as shown in **Fig. 8**.
4- An FTP file was setup with a total size of three files that were produced from the MIP model in the environmental LAB as shown in **Fig. 9**.

7. RESULTS AND DISCUSSION

The experimental results obtained from the OPNET simulator had relied on adjustment of the TCP parameters after passing the three files resulting from the MIP, i.e. Cipgher1, Cigher2 and KEK. These results had been exported in Ms Excel Sheet to check for correlation between these parameters. The results collected had relied on a permutation among TCP attributes in OPNET simulation and observing changes in the throughput. This procedure is illustrated in **Fig. 10.**

a. Window Scaling is disabled

Fig. 11 shows the comparison between the three congestion mechanisms; New Reno, Tahoe and SACK. The RWIN has changed its value and was increased. In this Figure the value of initial retransmission timeout (RTO) is fixed to 0.5 second. Slow start initial count is set to 1. Under that condition the SACK mechanism had achieved a maximum throughput of 27824 bit/s when the received window size value was 16KB.

Fig. 12 shows that the result when changing the condition of slow start initial count to 2 and the initial retransmission timeout (RTO) to 1 seconds using different values of RWIN with New Reno, Tahoe and SACK mechanisms. Under these conditions a high throughput of 28054bit/s is achieved when using SACK mechanism with RWIN of 8 KB.

b. Window Scaling is Enable

Fig. 13 displays the results when setting slow start initial count to 1 and initial retransmission timeout (RTO) to 0.5 seconds, using different values of RWIN with New Reno, Tahoe and SACK mechanisms. In this case a high throughput of 28152bit/s is achieved when using SACK mechanism with RWIN of 16 KB.

The same conditions are kept in **Fig. 14** where the slow start initial count is set to 1 and initial RTO is started with 2 second. In this case the Tahoe mechanism had achieved high throughput of 31398 bit/s when the RWIN was 64 KB.

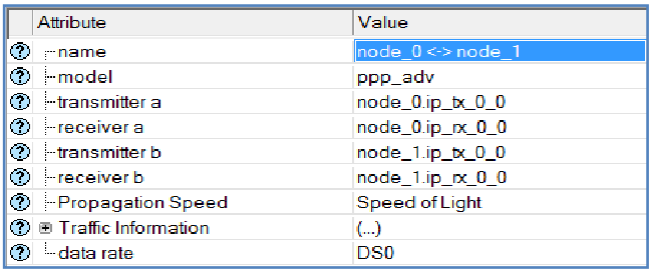


Fig. 7. The configuration to simulate the satellite link

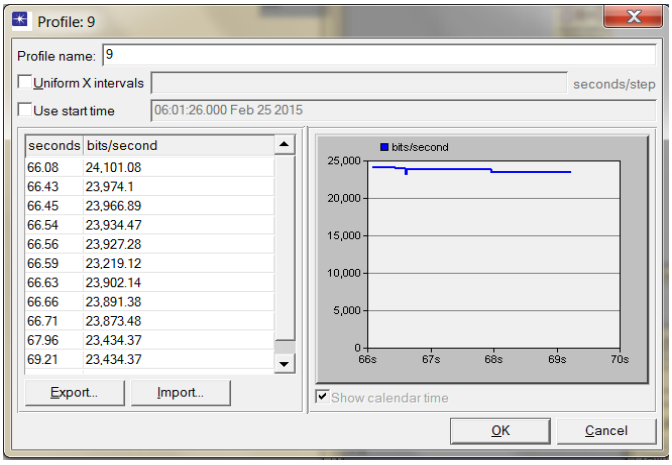


Fig. 8. Feeding the simulator with the transmission data rate behavior as in real environmental experiment

Attribute	Value
Command Mix (Get/Total)	100%
Inter-Request Time (seconds)	constant (1)
File Size (bytes)	constant (199075)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Fig. 9. Setup of the FTP file in the simulator

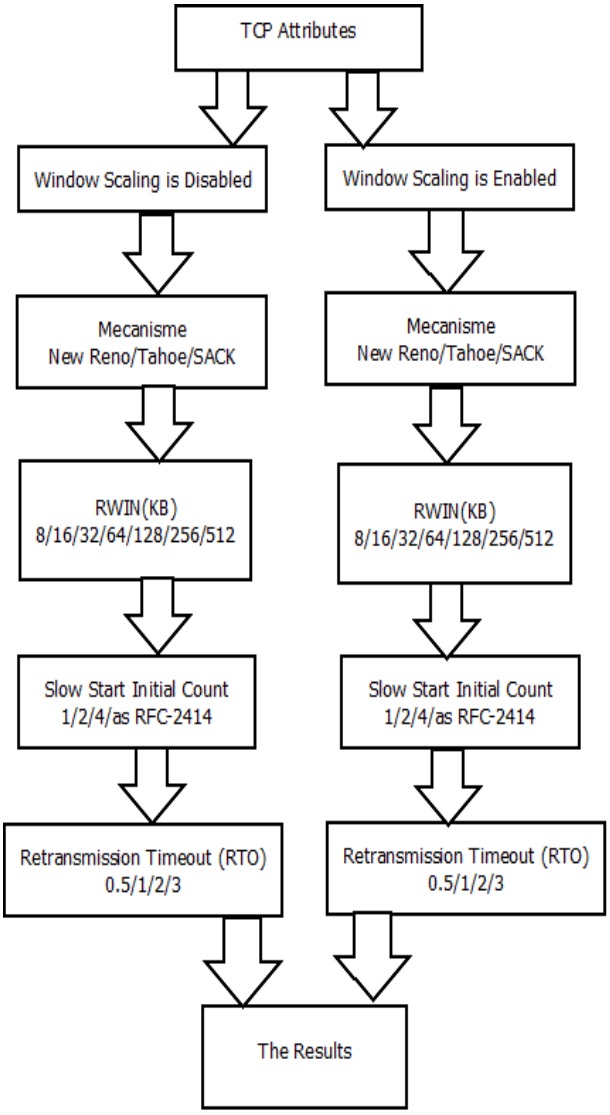
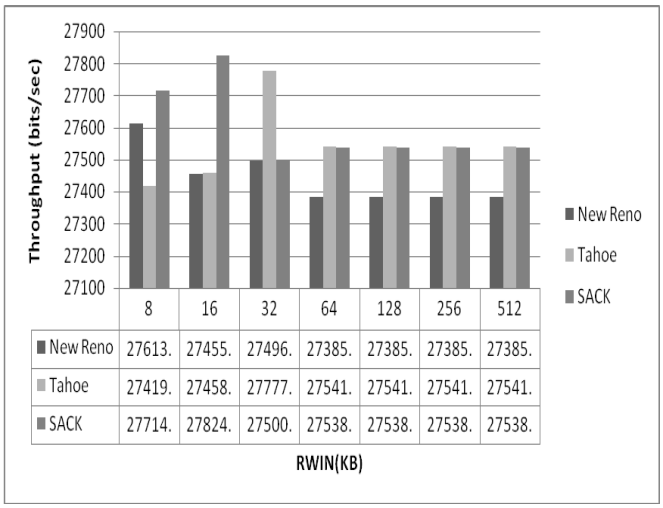
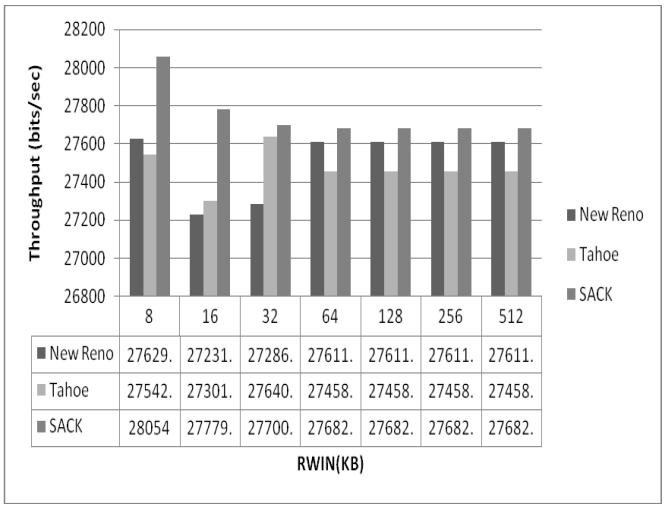


Fig. 10. Methodology of Collecting the Results



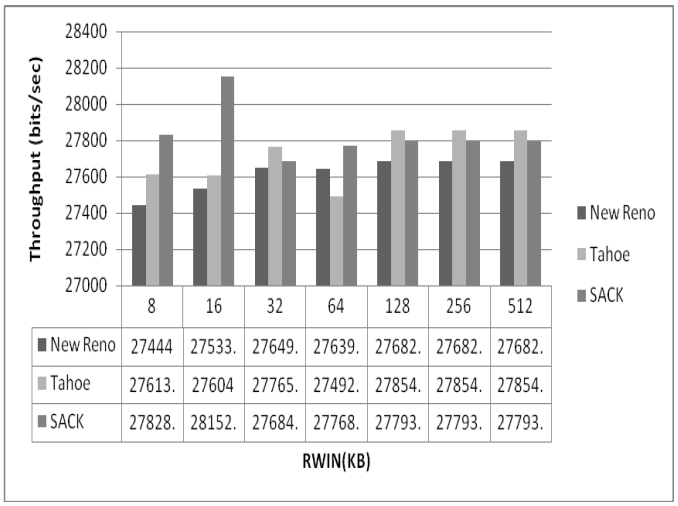
	8	16	32	64	128	256	512
New Reno	27613.	27455.	27496.	27385.	27385.	27385.	27385.
Tahoe	27419.	27458.	27777.	27541.	27541.	27541.	27541.
SACK	27714.	27824.	27500.	27538.	27538.	27538.	27538.

Fig. 11. Throughput when slow start initial count is 1 and initial RTO is 0.5 seconds



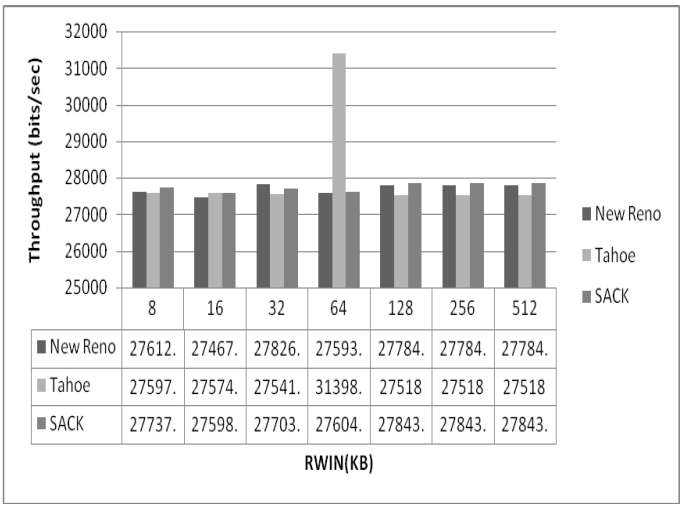
	8	16	32	64	128	256	512
New Reno	27629.	27231.	27286.	27611.	27611.	27611.	27611.
Tahoe	27542.	27301.	27640.	27458.	27458.	27458.	27458.
SACK	28054	27779.	27700.	27682.	27682.	27682.	27682.

Fig. 12. Throughput when slow start initial count is 2 and initial RTO is 1 second



	8	16	32	64	128	256	512
New Reno	27444	27533.	27649.	27639.	27682.	27682.	27682.
Tahoe	27613.	27604	27765.	27492.	27854.	27854.	27854.
SACK	27828.	28152.	27684.	27768.	27793.	27793.	27793.

Fig. 13. Throughput when slow start initial count is 1 and initial RTO is 0.5 second



	8	16	32	64	128	256	512
New Reno	27612.	27467.	27826.	27593.	27784.	27784.	27784.
Tahoe	27597.	27574.	27541.	31398.	27518	27518	27518
SACK	27737.	27598.	27703.	27604.	27843.	27843.	27843.

Fig. 14. Throughput when slow start initial count is 1 and initial RTO is 2 second

8. CONCLUSIONS

This paper has presented a cryptography protocol that is called MIP to enhance packets security being transferred through the satellite channel. It gave a solution to the man-in-the-middle attack problem in double hop topology. The paper suggested the use of a cipher block chaining mode to split the message and send hash function of the message instead of the traditional Interlock Protocol. The work presented here has proposed a model to do so. For enhancement of TCP performance it is suggested to adjust the TCP parameters using OPENET Model simulator. By doing so, the throughput had increased from 24101 bit/s in the environmental lab to 31398 bit/s after the TCP parameters had been adjusted.

REFERENCES

[1] Anil K. Maini and Varsha Agrawal, Satellite Technology Principles and Application, Second Edition 2011.

[2] https://en.wikipedia.org/wiki/Certificate_authority.

[3] Bruce Schneider, Applied Cryptography. Second Edition, January 1996.

[4] R. Rivest and A. Shamir, How to Expose an Eavesdropper, April 1984.

[5] D. W. Davies and W. L. Price, Security for Computer Networks. John Wiley & Sons, second ed., 1989.

[6] James F. Kurose and Keith W. Ross, Computer Networking, Pearson, 2003.

[7] Jing Peng, Improving TCP performance over long delay satellite links, July 2001.

[8] M. Allman, S. Floyd and C. Partridge, Increasing TCP's Initial Window, IETR RFC 2414, 1998.

[9] J. Postel, The TCP Maximum Segment Size and Related Topics, IETR RFC 879,1983.

[10] V. Paxson and M. Allman, Computing TCP's Retransmission Timer. IETF RFC 6298, 2000.

[11] W. Stevens, TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, IETR RFC 2581, 1997.

[12] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, TCP Selective Acknowledgment Options, IETR RFC 2018, 1996.