ML-IPSec+: An End to End Accelerated VPN for Satellite Links

H. Fereidooni*, A. Parichehreh, H. Taheri*, M. Mahramian***, B. Eliasi****

* Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran.
** Iran Telecommunication Research Center, PO Box 14155-3961, Tehran 14399, Iran.
***Informatics Services Corporation, Tehran, Iran.

Summary

TCP protocol has been designed for E2E data transfer in congested networks. TCP performance degrades in satellite links because of the inherent delay. Accelerating methods are used to enhance TCP performance over satellite links by employing Performance Enhancement Proxies (PEPs). However, providing a secure connection through the PEPs seems to be impossible. In this paper an appropriate method is proposed in order to provide an accelerated secure E2E connection. ML-IPSec+ improves available solutions of TCP performance enhancement over satellite links, while increases the E2E security level using the key exchange protocol.

Keywords: Key Exchange, Encryption, TCP Acceleration, Authentication, Performance Enhancement Proxy.

1. Introduction

TCP Acceleration is a series of techniques for achieving better throughput on Internet connections over satellite links, without modifying endpoint applications (Fig. 1). TCP performance enhancement proxies (PEPs) are effective tools to maximize satellite link efficiency. They improve the end-to-end performance of some communication protocols such as TCP. A TCP PEP may locally acknowledge the received data segments from sender or even retransmit the segments lost on the path between the TCP PEP and the receiving end system. This leaves the end systems unmodified and can overcome some problems with TCP window sizes on the end systems of satellite communications.

Manuscript received January 5, 2009
Manuscript revised January 20, 2009

PEP provides the user**:**

- Enhancement of the two most widely used Internet protocols (HTTP and TCP) via satellite links.
- Efficient usage of link bandwidth.
- Faster access to websites, efficient Internet browsing and file transfer.
- Congestion control.

Virtual Private Networks (VPNs) provide the user with secure duplex connection channels. VPN tunnels encrypt user information at both ends to ensure secrecy and authentication. Therefore, sender and receiver sides, before any data transfer, need to exchange encryption keys [1] (Fig. 2). Protocols such as IPsec interfere with PEPs performance on implementing VPNs over satellite links [2], [3]. There are some techniques to set up a secure VPN over satellite links [4].

- Trusted PEPs:

This solution is based on trusting PEPs. PEPs require accessing TCP and IP headers of the packets for acceleration, so PEPs can change packets inside IPsec channel. In this method the users should trust the PEPs.

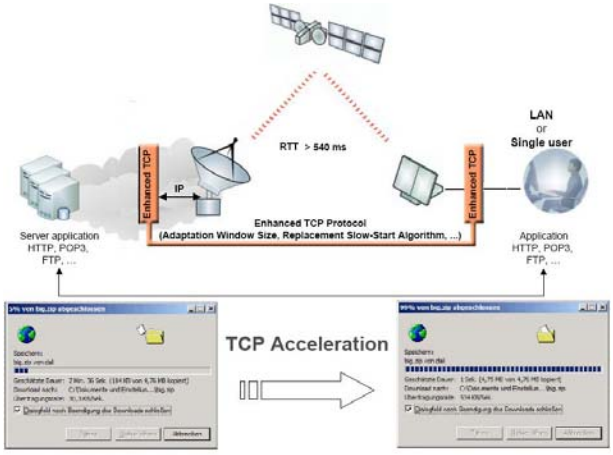

Figure 1.TCP Acceleration

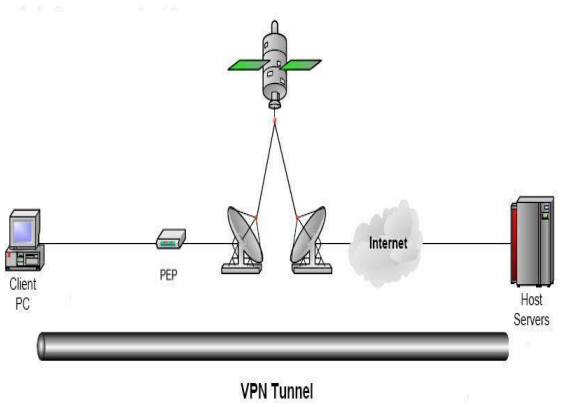

Figure 2. E2E VPN Connection

- Application layer Security Protocols:

A relatively easy and cost effective way to implement VPN over satellite is to encrypt just the TCP payload, leaving the TCP or IP headers unencrypted. However, application layer security protocols (SSL/TLS) provide good level of security for small businesses, conflict with an intensive security policy.

- Changing IPsec to make the header accessible:

IPsec can be modified to make headers accessible to TCP accelerators. ESP protocol, which is responsible for encryption of packets, changes in a way that IP addresses and port numbers will not be encrypted on both sides.

- Multilayer security (ML-IPsec):

This algorithm divides each IP packet into different sections, and encrypts each section independently. In this method, PEPs have only header encryption key, thereby they do not have access to the payload. So, we can provide an end to end secure connection. Among previous studies, this method is more acceptable from the viewpoint of security (Fig. 3).

The proposed method i.e. ML-IPsec, replaces IPsec single layer model with a multilayer security model This method is based on dividing IP packets into several zones, using a specific security pattern for each zone. Thus, the PEPs can access limited parts of IP packet. Authentication and decryption keys for each zone are different. PEP, whose access is granted to limited parts of the packet by ML-IPsec, can only decrypt its own part, and after applying changes, will encrypt it again. ML-IPsec has encrypted TCP header and application data header part in every IP packet separately and reveals decrypting key only for final sender and receiver. TCP header decryption key will be accessible for some of trusted PEPs, thus the whole process assures end to end security [5].

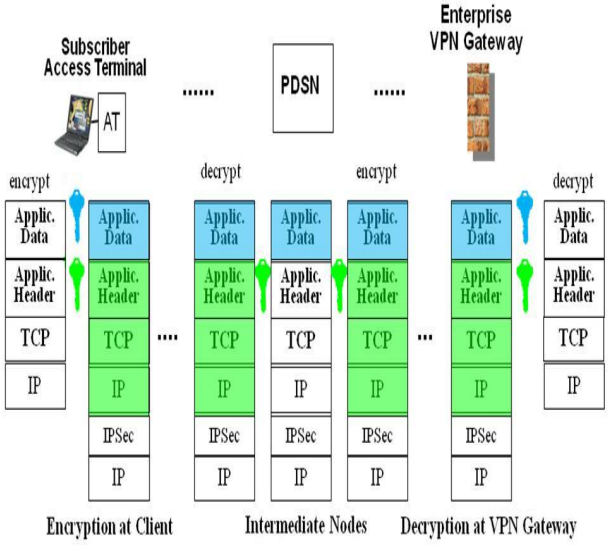

Figure 3.Multilayer IP Security Model

One of the most important features of the proposed method is limiting PEP's access to different parts of IP packet along the path. Here, we can realize the importance of using IKE protocol for exchanging public key and authentication of each PEP. Using IKE protocol, security of key exchanges is extremely improved. We use a generation of ML-IPsec to extend IKE protocol in order to support multilayer IP security. As mentioned before, to increase security in ML-IPsec exchanges, IKE protocol is used. The proposed algorithm is called "**ML-IPsec+".**

The following section explains the key exchange protocol, authentication method and ML-IPSec+ method. Section 3 applies some of the frequent attacks to the secure connections and presents the resistance of the ML-IPSec+ against the hackers and Section 4 concludes the paper.

2. Key Exchange Protocol

IKE is a protocol for distribution of the key in IPsec, which is responsible for encryption key management in IPsec [6]. Although IPsec assumes that there is an agreement for the data security, it can do nothing for key management. IPsec uses SA (security association) for key management. SA determines how two or more stations communicate with the proper key(s). To set up SA, both sides must be authenticated (Fig. 4).

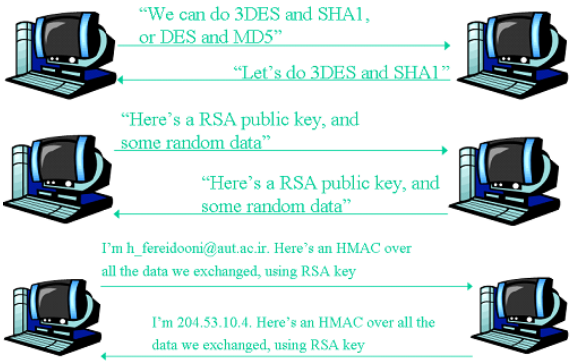

Figure4. Negotiation between sender and receiver (Main mode)

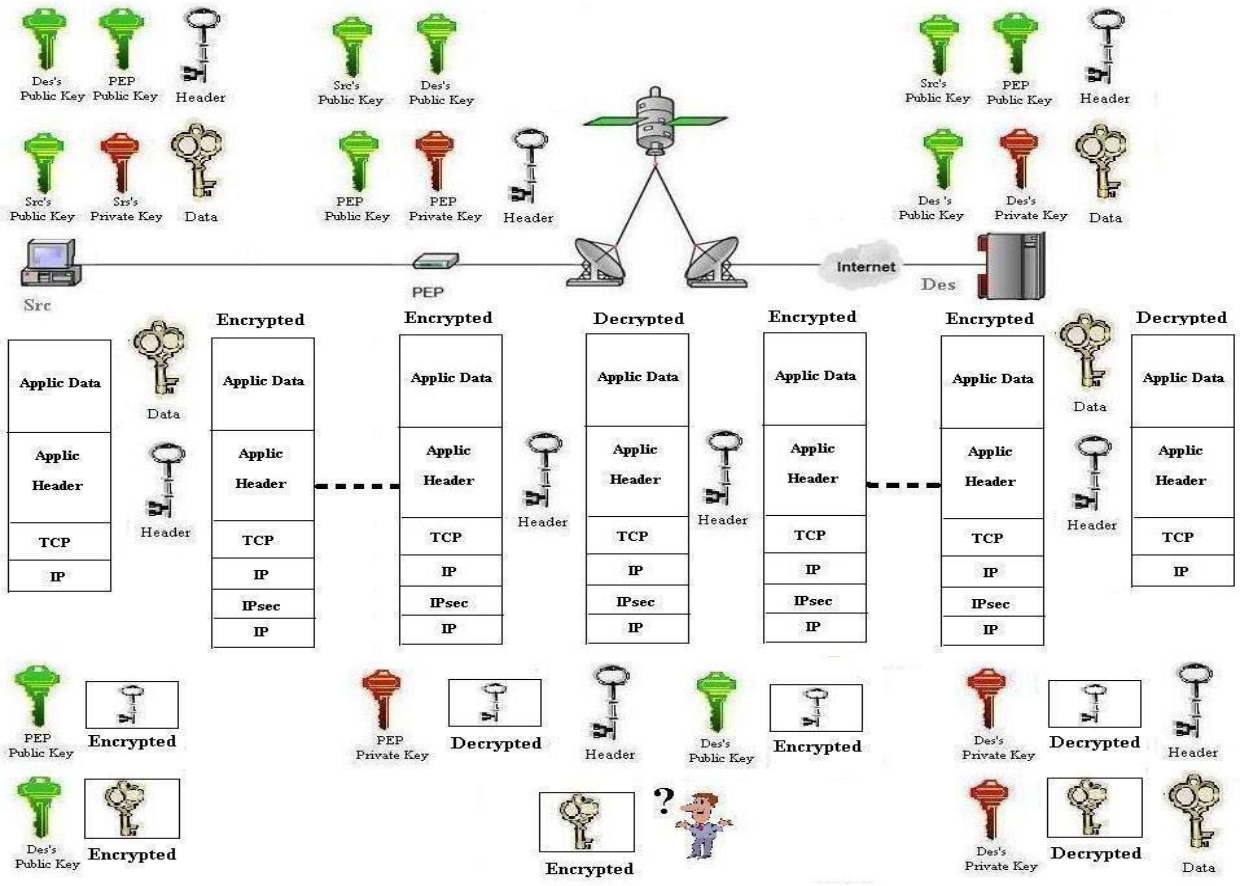

Figure5. Key distribution in ML-IPSec+

2.1. Authentication Methods

Following methods are used for authentication:

- Pre shared keys: a key is installed on both sides. IKE generates a hash number by the key and sends it to the destination. If both sides are capable of generating the same number, they will both have it.
- Public key encryption: each side generates a random number and after encrypting it by other side's public key, sends it to the destination. If the destination is capable of decrypting the number by its private key and retransmits its encrypted form using the source's public key to it, then the connection is authenticated.
- Digital signature: in this method each of the two sides signs one string of data, and sends it to the destination. To provide a secure connection, both sides must first agree on a key [7].

In this paper the public key method is utilized because of its proper performance (Fig. 5).

2.2. ML-IPsec+

Before data exchange, both ends should authenticate the other side. For key exchange, the following algorithm is proposed. Our notation is shown in Table 1.

Table.1

S,P,R	System principals (sender , PEP & receiver)
Hdr key	Symmetric (secret-key) encryption of header with it
Data key	Symmetric (secret-key) encryption of data with it
Pub	Asymmetric (pub-key) encryption of data and header keys with public key
Pri	Asymmetric (private-key) encryption of data and header keys with private key

The data sender is responsible for providing encryption and decryption key of packet header for PEP, as well as, encryption and decryption key of both data and header for final receiver. In order for the sender to be informed of PEP and receiver real identity, sender encrypts header's decryption key with the PEP public key, and sends it for PEP. As a result, only PEP can decrypt it with its own private key. After the key transfer phase, sender computes hash of the sent key and encrypts it with its own private key and sends it to the PEP.

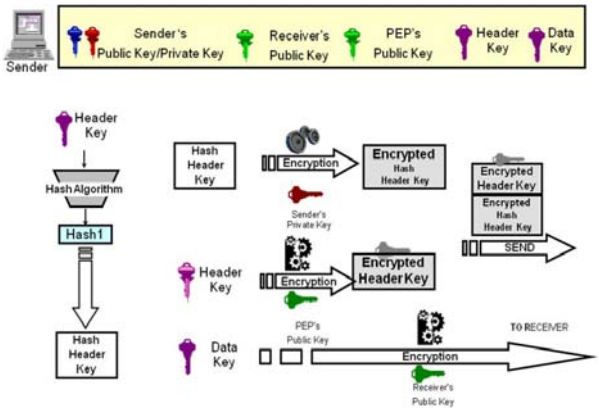

Figure6.a. Key exchange and authentication between sender, PEP

S → ***R****: Encrypt {data key, Pub (R)}*
S → ***R****: Encrypt {hash (data key) ,Pri(S)}*
S → ***R****: Encrypt {hrd key,Pub (P)}*
S → ***R****: Encrypt {hash (hdr key), Pri (S)}*

PEP can decrypt the hash, because it has the sender's public key. The PEP decrypts the received key by its own private key.

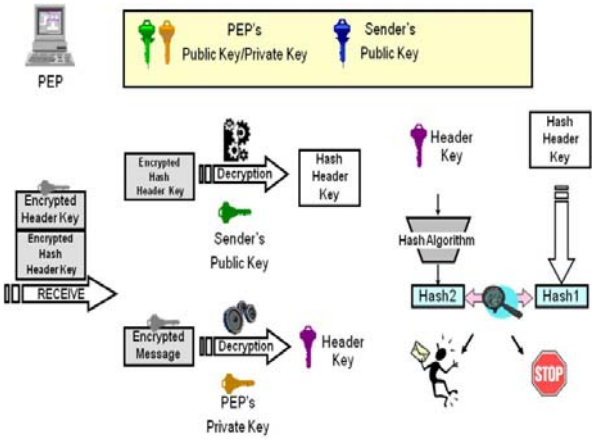

Figure 6.b. Key exchange and authentication between sender, PEP

decrypt *{encrypt (hdr key, Pub (P)), Pri (P)}*
= hdr key
decrypt *{encrypt (hash (hdr key), Pub (P)), Pri (P)}*
= hash (hdr key)
Note*:"Encrypting with a private key can work as signin:.*
Sign (massage) = Encrypt {massage, private key}"

Then, based on the algorithm on which the sender and PEP have agreed, the PEP generates a hash. The PEP compares its own hash with the hash it has received from the sender.

auth *{sign (hash (hdr key)}*

PEP realizes that the key is due to the sender, if and only if the two hashes are similar. Since the private key of each station represents its real identity and is not available to the other stations, the received header key is guaranteed to be from the recognized station. The PEP performs acceleration procedures after decryption the packet's header, which is the only part of the packet that PEP can decrypt. Then by using the same key (secret header key), encrypts the packet header and after encryption of secret key by final receiver's public key, sends it to the receiver.

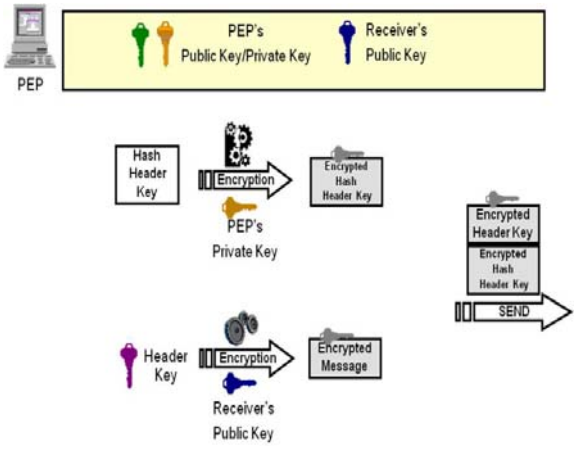

Figure7.a. Key exchange and authentication between PEP, receiver

P $\rightarrow$ ***R***: *Encrypt {hdr key, Pub (R)}*
P $\rightarrow$ ***R***: *Encrypt {hash (hdr key), Pri (P)}*

Scrambled script (hash) of the header encryption key is encrypted by the PEP's private key, and is sent to the final receiver. The PEP's private key represents the PEP's actual identity. Encrypted and scrambled script is opened by the PEP's public key which is received by the receiver and is compared with scrambled script (hash) which the receiver makes using the received key. If both keys are the same, the PEP's identity is authenticated.

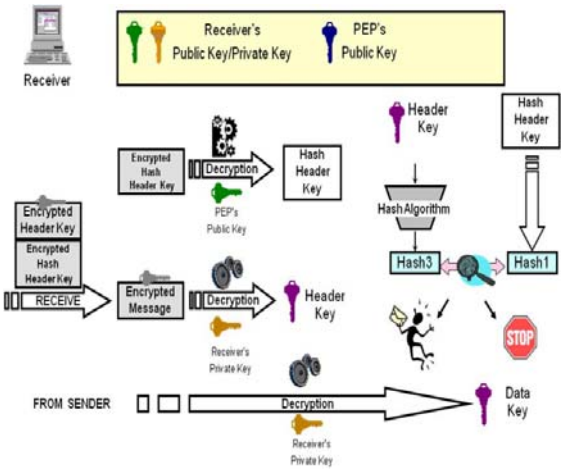

Figure 7.b. Key exchange and authentication between PEP, receiver

decrypt *{encrypt* ***(****hdr key, Pub (R)****)****, Pri (R) }*
= hdr key
decrypt *{encrypt* ***(****data key, Pub (R)****)****, Pri (R)}*
= data key
decrypt *{encrypt* ***(*** *hash (hdr key), Pri (P)* ***)****, Pub (P) }*
=hash (hdr key)
auth *{sign (hash (hdr key)}*

decrypt *{encrypt* ***(*** *hash (data key), Pri (S)* ***)****, Pub (S)}*
= hash (data key)
auth *{sign (hash (data key)}*

Data decryption key is encrypted by the sender using receiver's public key, and is sent to the receiver. Final receiver decrypts message by its own private key, considering the fact that identity authentication is also performed between the final receiver and the sender (Fig. 6).

3. Proof of the concept

After discussion on the "ML-IPsec+" method we investigate some kinds of conventional attacks to the network and study the security level of "ML-IPsec+" proving that the utilized algorithm is a secure method against such attacks.

- Spoofing Attack:

An attempt by someone or something to masquerade as someone else usually considered as an access attack. The popular spoofing attack today is IP spoofing. The goal of IP spoofing is to make the data look as if it came from a trusted host when it didn't.

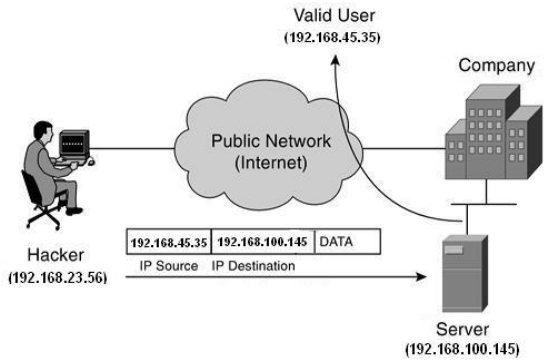

Figure 8 .IP Spoofing

The Internet Protocol (IP) portion of TCP/IP carries the information describing where the packet is coming from and where it is going. This information consists of IP addresses. In IP spoofing, an attacker pretends to be sending data from an IP address other than his own. TCP/IP assumes that the source address on any IP packet it receives is the same IP address as the system that actually sent the packet (which is a vulnerability of TCP/IP in that it incorporates no authentication). Many higher level protocols and applications also make this assumption, so anyone able to fake or forge the source address of an IP packet could be authorized as a user, sometimes with special privileges. This practice is called "spoofing" an address. There are two difficulties in this spoofing technique. The first is that all communication is likely to be one way. The remote host will send all replies to the spoofed source address, not to the host actually doing the spoofing. Thus, an attacker using IP spoofing is unlikely to see output from the remote system unless he has some method of eavesdropping on the network between the other two hosts. The second disadvantage, from the spoofer's point of view, is that an attacker needs to use the correct sequence numbers if he plans on establishing a TCP connection with the compromised host. Another way to do IP spoofing makes use of an IP option called "source routing." Source routing allows the originating host to specify the path (route) that the receiver should use to reply. Any attacker can take advantage of this by specifying a route that bypasses the real host and instead directs replies to a path it can monitor (probably to itself). Although simple, an attack using source routing may be unsuccessful, because most routers now are configured to drop

packets with source routing enable (Figure7). In “ML-IPsec+” , if the attacker tends to fake the IP address of the sender, he can not pass the authentication phase because he does not know the private key of the sender for digital signature. Also, the secret keys which have been provided for the receiver by “ML-IPsec+” are not accessible to the attacker, leading to an ineffective attack.[8,9].

- Connection or Session Hijacking

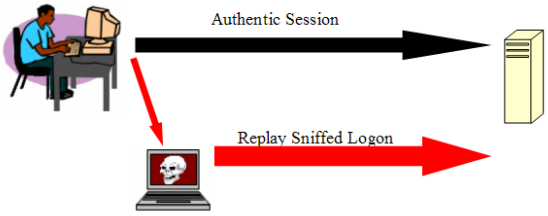

Figure9.a.Attacker hijacks the session to bypass the authentication phase

A way to accomplish IP spoofing is for a host to insert itself in the middle of a connection between two other hosts. This is called “connection hijacking” or “session hijacking.” IP spoofing alone may not be able to bypass additional security, such as an authentication measure that has been added or enforced on the operating system, but session hijacking allows an attacker to bypass the authentication phase and proceed the connection between the two hosts, and then seize control of the connection. Session hijacking exploits a desynchronized state in TCP communication. (Figure8.a) Hijacking is taking over an already established TCP session and injecting the attacker's packets into that stream so that his commands are processed as the authentic owner of the session. (Figure8.b)

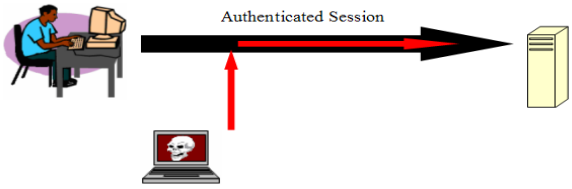

Figure 9.b.Session hijacking

To complete a hijack you must perform 3 actions:

- Monitor or track a session
- Desynchronize the session
- Inject your own commands

To monitor a session, you simply sniff the traffic. We achieve the de-synchronization of a session by ‘Packet Sequence Prediction’. (Figure8.c)

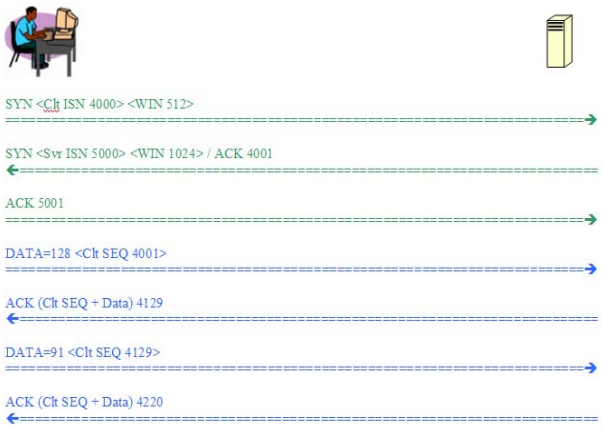

Figure 9.c.Desynchronizing the client by packet sequence prediction

We now know the next expected sequence number. If we transmit a packet with the expected sequence number before the client, we will desynchronize the connection; basically we will bump the server up by one increment.

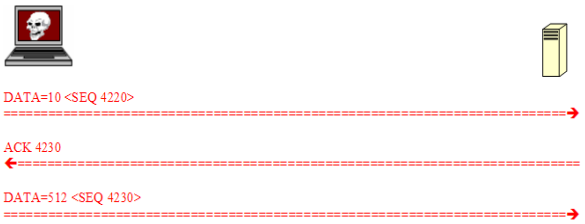

Figure 9.d. Client can not continue the connection

When the real client sends the next packet, the server treats it as a resent packet as it has already received that SEQ number. So, now the client is unable to communicate with the server, the hacker is still able to communicate as he knows the correct sequence number. (Figure8.d)

Figure 9.e.Session hijacking

The attacker is not able to change the encrypted information, since it dose not have access to the symmetric secret keys even if it can guess the sequence numbers correctly. In case of sending fake information with valid sequence numbers during the

authentication procedure, the receiver will notice that the session is not authorized and reject it, because the attacker can not access the secret key and the hash massage authentication code (HMAC=Hash(data + secret key)) is also invalid, thereby session hijacking will not occur. (Figure8.e)[9].

- Man-in-the-Middle Attack (MITM)

The man-in-the-middle is an attack in which an intruder is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised. The attacker must be able to observe and intercept messages going between the two victims. MITM attacks occur due to the lack of authentication, or weak authentication being performed between the two legitimate parties involved in a transaction or communications session. (Figure9). The man-in-the-middle software may be recording information for someone to view later, altering it, or in some other way compromising the security of your system and session. In recent years, the threat of man-in-the-middle attacks on wireless networks has increased. Because it's no longer necessary to connect to the wire, an attacker can be outside the building intercepting packets, altering them, and sending them on.

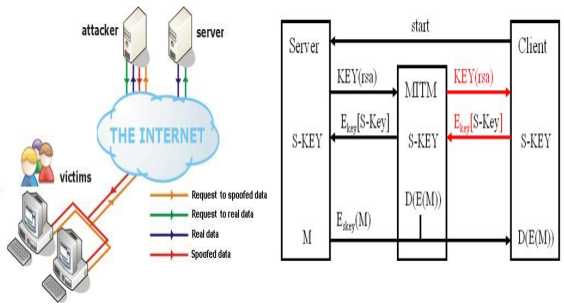

Figure10. Man in the middle (MITM)

In the "ML-IPsec+" method, the RSA public key algorithm is used, which utilizes asymmetric public and private keys for encryption and decryption. One of the most important properties of the RSA algorithm is that the private key could not be guessed from the public key. Since the private key is completely secret, the attacker is not able to share the key with the communication parties. In case of abusing the public key and sending fake information in authentication step, the attacker will be identified due to unavailability of the private key and the attack will not be effective[10].

- Replay Attacks

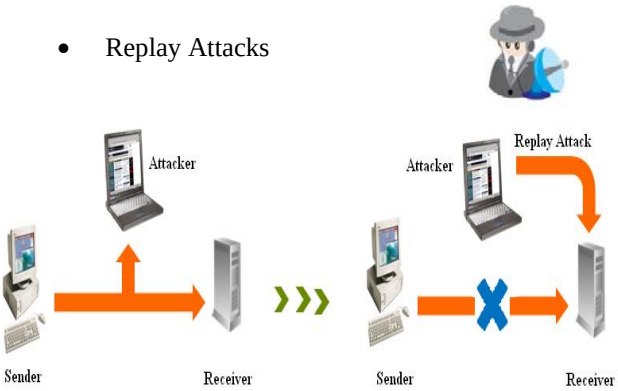

Figure 11.Replay attack

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. (Figure10). Many solutions provide the mechanism of encrypting the ongoing data exchange packets between two peers. Even when the packets are encrypted, the users are still prone to another intrusion, replay attack. Where the attacker uses pre-validated packets and sends them to one of the users to confuse and disrupt the communication. The receipt of duplicate and authenticated IP packets may disrupt service in some way or may have some other undesired consequence. Replay attack occurs when information is captured over a network. In a distributed environment, logon and password information is sent between the client and the authentication system. The attacker can capture this information and replay it again later. This is the primary reason that most certificates contain a unique session identifier and a time stamp, if the certificate has expired, it will be rejected. The sequence number field is designed to thwart such attacks. When a new SA is established, the sender initializes a sequence number counter to 0, each time that a packet is sent on this SA, the sender increments the counter and places the value in the sequence number field. Thus, the first value to be used is 1. If the limit of $2^{32}-1$ is reached, the sender should terminate this SA, and negotiate a new SA with a new secret key. To prevent replay attacks, whenever the sequence number reaches $2^{32}-1$, the session is disconnected and another session is made with the new secret key. To do this, the symmetric keys of encryption and decryption are swept and sent for the receiver by the public key. All of the steps of authentication and digital signature are performed again therefore existence of the duplicate packets do not cause disturbance since the encryption and decryption keys have been changed and the receiver can detect replay attack [11].

4. Conclusions

Key exchange and distribution algorithm not only authenticate the actual identity of the sender and receiver, but also enhance the security criteria of the transfer. ML-IPsec+ algorithm does not require saving encryption and decryption keys for a long period. This fact results in a lower probability of detection of the key by the others and helps us change the key periodically. Also, it provides an end to end accelerated secure connection over satellite links. The algorithm enhances the data transfer security, and provides a more secure connection. Some widespread attacks are applied to the ML-IPSec+ to prove its resistance against hackers.

Acknowledgments

The authors are thankful of the financial support provided by Iran Telecommunication Research Center (ITRC).

References:

[1] E. Olechna and P. Feighery, "Virtual Private Network Issue Using satellite Based Networks", IEEE Military communication conference, vol.2, pp.785-789, 2001.

[2] "An Introduction to IP Security (IPSec) Encryption" Available at: http://www.cisco.com/application/pdf/paws/16439/IPSECpart1.pdf

[3] "High Performance VPN solutions Over satellite Networks," White paper, Encore Networks, October, 2004.

[4] D. Demirel, F. Alagoz, and M. Ufuk, "IPsec over Satellite Links: A New Flow Identification Method," 7th IEEE International Symposium on Computer Network, pp.140-145, 2006

[5] Y. Zhang, "A Multilayer IP Security Protocol for TCP Performance Enhancement in Wireless Network," IEEE Journal on Selected Areas in Communications, vol.22, pp.767-776, 2004.

[6] "Internet Security Association and Key Management Protocol (ISAKMP)" Available: http://tools.ietf.org/html/rfc2408

[7] H. Beker and F. Piper, "Cipher systems to protection of communications," Distributed: VAN NOSTRAND REMINHOLD (U.K.), Northwood Publications,1982.

[8] Internet Security White Paper, IT Security Series. June2007.

[9] "IP Spoofing Attacks and Hijacked Terminal Connections". Available at: http://www.cert.org/advisories/

[10] R. Zakeri, R. Jalili, H. R. Shahriari, "Using Description Logics for Man In The Middle Attack Analysis", 11th International CSI Computer Conference (CSICC'2006), School of Computer Science, IPM, Jan. 24-26, 2006, Tehran, Iran.

[11] M. Barbeau, "Mobile and Wireless Network Security," Tutorial Notes, School of Computer Science Carleton University.

Hossein Fereidooni received BSC. and MSC. degrees from Amirkabir University (Tehran Polytechnique) in 2007 & 2009 respectively. After working as a teacher assitant from 2007 in dept. of Biomedical and Electrical Engineering, the Amirkabir university of technology, he has work in Reaserch Complex of Amirkabir of University of Technology since 2008. His research interest includes Satellite communication networks, Wireless networks, RF MEMS and their application in medicine and Wireless networks and network Security.

Ailin Parichehreh received BSC from Amirkabir Univ. in 1999 & and MSC from Azad Univ. in 2003. Since 2002 she has worked in Iran Telecomm. Research Center (ITRC), where she is currently a member of the Satellite Communications Research Group. Her current research interests include satellite communication networks, wireless networks security, and broadband communications by HAPs.

Dr. Hassan Taheri received the BSC. degrees from Amirkabir University of Technology, MSC. and PhD. from University of Manchester Institute of Science and Technology in 1975, 1978 and 1988 respectively; all in Electrical Engineering. He is now an associate professor in the Department of Electrical Engineering at the Amirkabir University of Technology (Tehran Polytechnique).

Dr. Mehran Mahramian received the BSC. and MSC degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 1995 and 1997 respectively. Since 1997, he is with Informatics Services Corp. to study and implement broadband satellite communication equipments. He is graduated from Amirkabir University of Technology (Tehran Polytechnique) for the PhD degree in Electrical Engineering.

Behrooz Eliasi received BSC. from Ferdosi Univ. in 1989 & and MSC. degrees in 1996 from Khajeh Nasir Toosi Univ. His thesis has been on MAC sublayer protocols in Sat. Communicatiuons. Until then, he has worked in Iran Telecomm. Research Center (ITRC), experiencing on different projects in satellite communicions area. His research interests include satellite communication networks and security, AMC and ARQ wireless networks and broadband communications by HAPs.