Security Issues in Internet Protocols over Satellite Links*

Guevara Noubir & Laurent von Allmen
Real-Time Software and Networking Group
CSEM SA, Neuchâtel, Switzerland
guevara.noubir@csem.ch, laurent.vonallmen@csem.ch

Abstract

Security is an important issue in IP over satellite, since an attacker can easily intercept such communication and can even corrupt the transmitted data [Noub98]. In the first part of the paper we address the implications of optimizing the transport control protocol (TCP) on the security services provided by IPSec protocol suite. We provide a set of rules for optimizing TCP without interfering with IPsec. In the second part of the paper, we address IP multicast security issues. We also introduce an efficient key distribution algorithm that can handle a large dynamic group.

1 Introduction

Since TCP was optimized for low delay symmetric lines, it shows a loss of performance when the IP packets are transmitted over satellite. This is mainly due to the high round trip time (RTT) of such links and to the asymmetric nature of the link bandwidth. Several propositions have been made to optimize TCP for satellite. Optimization is done using techniques such as: selective acknowledgment (SACK), random early detection (RED), acknowledgment filtering, header compression or acknowledgment rate reduction. However, the proposed techniques may interfere with the use of IPsec to protect the IP packets. For example optimization techniques involving the intermediate routers and that require an access to part of the TCP PDU can not be used. We show how the proposed techniques affect the security services of IPsec (i.e., confidentiality, authentication and integrity). We also derive several rules for optimizing TCP without interfering with IPsec services. These rules take into account the point were optimization technique is applied (e.g., end hosts, intermediate routers), the type of access it requires to the TCP PDU (e.g., read or write), and the security service required by the application.

Multicast communication provides an efficient way for delivering data to multiple receivers. Several applications may benefit from this type of communication such as data distribution (e.g., stock market information broadcasting, news distribution), video broadcasting, collaborative work, distributed simulation, distributed games. Satellite links provide an efficient way for enabling multicast communications for large groups. Since many multicast applications are for paying users, such communication has to be secured to allow only the group members to receive the information. One of the most important issues in multicast communication is the distribution of keys. Only the current members of the group should be able to decrypt the information (nor the members who left the group neither the members who have not joined yet). Thus, the group key has to be changed at each membership change. This is a particularly difficult task for large dynamic groups [KA97, Mitt97, HC97] because known key distribution algorithms do not scale. We will present an algorithm that requires only an O(log |M|) (where |M| is the group number of members) message size at each group membership change (join or leave) [Noub98, Noub99] while the most efficient known algorithms require an O(|M|/N) message size (where N is the number of security agents) [Mitt97]. In addition, it can be implemented in a centralized manner (thus reducing the number of agents that can be attacked), and can efficiently adapt to groups that regularly increase their size.

2 Securing IP over satellites

In this section, we will discuss IP over satellite security requirements and show how IPsec could provide such security. Finally, we will discuss how the optimization techniques of TCP discussed in

* This work was done within the project “Optimization of the Internet Protocols over Satellite”. This project was financed by the European Space Agency.

[Hurl98] may influence the IPsec implementation. We will also provide basic rules for optimizing TCP without interfering with IPsec.

2.1 Security services required for IP over satellite

Satellite communications are specially sensitive to security attacks, because the transmitted data can be easily intercepted and corrupted. Thus, it is very important to provide strong security mechanisms:

- **Confidentiality**: this service protects data from passive attacks. It protects against unauthorized release of message content. It may also provide a protection against traffic flow analysis (e.g., source, destination, frequency, length). This service has to be provided (unless if the user does not need any communication privacy e.g., free information broadcasting). This protection is needed because it is very easy for an attacker to intercept satellite communications.
- **Authentication**: this service guarantees that the communication is authentic. It assures the recipient of a message that the message is from the source that it claims to be from. It is necessary to have a strong authentication mechanism since it is very easy to impersonate a user on a wireless communication and that the IP address fields can not be trusted.
- **Integrity**: this service assures that the message (or the protected part of it) is received as sent. Packet integrity has to be cryptographically protected. An attacker can easily modify packets and create new packets if there is integrity protection. This service have to be provided.
- **Non-repudiation**: this service prevents either sender or receiver from denying a transmitted message. This service may be necessary for some applications.
- **Access control**: this service allows to limit access to host systems and applications via communications links. This service requires that the authentication service be available.
- **Key management and exchange**: this service allows to negotiate security keys between communicating entities. While the other security services can be implement in a similar manner for unicast and multicast communications, the key management service is much harder to extend from unicast to multicast.

2.2 IPsec for securing IP over satellites

IPsec protocols are designed for providing Authentication, Integrity, Confidentiality and non-repudiations. A limited protection against traffic flow analysis can also be achieved. These protocols can be used to secure IP over satellite links. In the next section we will discuss the implications of modifying TCP on IPsec.

2.3 Implications of TCP optimization on security

In this section we briefly analyze the techniques proposed to optimize TCP for satellite communications [Hurl98].

2.3.1 Congestion control optimization

All the techniques used for optimizing congestion control such as selective acknowledgment (SACK), random early detection (RED) or slow start acceleration act at TCP level. Furthermore, these techniques are applied at the end hosts or there is no need to decrypt the TCP data (RED randomly drops packets basing its decision on the buffer fullness). Thus, IPsec services can be implemented independently without restriction.

2.3.2 Network asymmetry

In this section we analyze the techniques proposed for optimizing TCP in the case of network asymmetry. We will show that these techniques are not inconsistent with IPsec security protocols.

2.3.2.1 Acknowledgment Filtering

The ACK filtering technique uses the cumulative nature of TCP acknowledgments. At each arrival of an ACK on a queue, the previous messages of the same connection are re-processed (e.g., removed). The potential problem with this technique, is that if the message is encrypted it cannot be identified as an ACK message (nor can the connection be identified). Since this technique is proposed for the receiver queue in the reverse path, the ACK is still accessible to the filtering entity. This is because, the ACK packet have not yet been processed by the security entity, which is located at the IP level.

2.3.2.2 Header compression

This technique reduces the header size of the acknowledgments. Since this technique is applied at the TCP level, it has no implication on the security aspects.

2.3.2.3 ACK rate reduction

This technique delays the ACK messages for a configurable period of time. It is executed at TCP level and has no implication on the security aspects.

2.3.3 Discrimination against long RTTs

The proposed solutions for handling the discrimination against connections with large round trip times are executed at TCP level between the communicating hosts. These solutions involve changing the congestion avoidance algorithm. Because these algorithms are executed between the end hosts at TCP level there is no implications for the security aspects.

2.4 Optimizing TCP without interfering with IPsec

Since the TCP packet transmitted over IP will certainly be encrypted we can deduce some basic rules on which type of TCP optimization techniques can be used without interfering with IPsec security protocols:

- If the optimization technique is used between the end hosts at the TCP level and does not involve the intermediate routers, it can be used with all IPsec security services (e.g., slow start acceleration, ACK filtering).
- If the optimization technique involves the intermediate routers but does not require access to the TCP data encapsulated by the IP protocol, it can be used with all IPsec security services (e.g., RED).
- If the optimization technique involves the intermediate routers and requires read access to the TCP data encapsulated in IP datagram, the IPsec ESP security service cannot be used. These techniques are not recommended for a communication over satellite, since confidentiality is an important service (such communication can be easily intercepted).
- If the optimization technique involves the intermediate routers and requires write access to the TCP data encapsulated in IP datagram, the IPsec security services (confidentiality, authentication and integrity) cannot be used. These techniques are not recommended for a communication over satellite, since it is very sensitive to security attacks.

3 Centralized hierarchical group key distribution

3.1 Key distribution scheme overview

This scheme is based on a tree-hierarchy of keys such that: when a member M_i leaves the group it is possible to change all the keys he had without M_i being able to get the new ones [Noub98]. The advantage of the hierarchy of keys is that this change of key requires only 2*log(|M|)-1 elementary messages. A similar version of this algorithm was independently discovered in [MTU98].

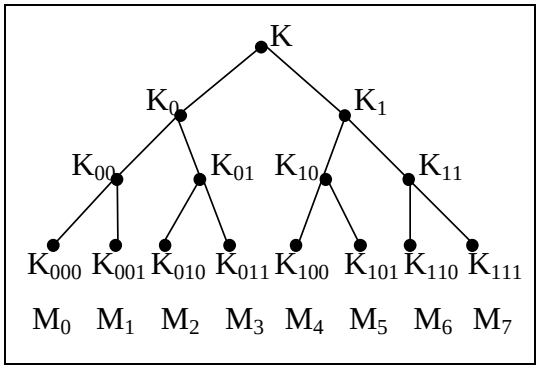

Figure 1. Hierarchy of keys. The leaf keys are the secret keys of the members. *K* is the group key and every group member have all the keys in the path from the root to its node (which constitutes log(|*M*|-1)+1 keys).

The set KS_i of keys given to the group member M_i is defined as follows: $KS_i = \{K_{m_l \ldots m_j} | 0 \le j \le l\} \cup \{K\}$ where l = log|M|-1, $m_l \ldots m_0$ is the binary representation of i (e.g., |M|=8, l = 2, and for member M_2 the set $KS_2=\{K_{010}, K_{01}, K_0, K\}$). The key K on the root node is the group key. The keys ($K_{ml \ldots m0}$) located on the leaves are the unique keys shared between the group controller and the group member.

3.2 Member join key update

When a member wants to join the group, the group controller sends him the set KS_i of keys. This set is constituted of all the keys located on the nodes

connecting the tree-root to the member leave. The group member already have the unique key $K_{Mi} = K_{ml\ldots m0}$ shared with the group controller. The old group key is noted K and the new one is noted K'. In this section we assume that the tree has enough leaves to accept the joining member. In section 3.5, we will show how the tree can be extended to add one more hierarchy level to double the number of its leaves.

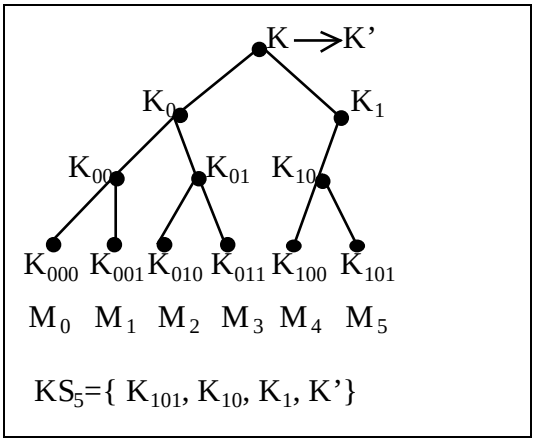

Figure 2. When a new member M_5 joins the group: 1) the group key K is changed into a new key K' and broadcasted to the group members. This message is protected by encrypting it using the old group key K. Then the set KS_5-{$K_{101} = K_{M5}$} is sent to M_5 after encrypting it using the key K_{M5} shared only by M_5 and *GC*.

Algorithm 1 ***Join***(M_i);

GC_send($E_K(K')$);
/* broadcasted to the group */
GC_send($E_{KMi}(KS_i-\{K_{Mi}\})$);
/* message for M_i ; K_{Mi} is not sent */
/* because M_i already have it */

3.3 Member leave key update

When a group member M_i leaves the group, the whole set of keys in KS_i have to be cancelled and changed to new ones. The principle of the algorithm is to change the keys starting by tree leaves (lower nodes) and then going higher in the tree hierarchy ($K_{ml\ldots mj}$ starting by j=1 and increasing until l).

The key $K_i = K_{ml\ldots m0}$, is cancelled. The key $K_{ml\ldots m1}$is used only by the neighbor of M_i ($M_{ml\ldots\overline{m0}}$). Thus it can be securely transmitted to $M_{ml\ldots\overline{m0}}$ using the key $K_{ml\ldots\overline{m0}}$. By induction we can assume that to change the key $K_{ml\ldots mj}$ we have already changed the key $K_{m_l\ldots m_{j+1}}$. Since, all the members who need to know the new key $K_{m_l\ldots mj}$ already have $K_{m_l\ldots m_{j+1}}$ or $K_{m_l\ldots\overline{m_{j+1}}}$ (which is not owned by M_i), then transmitting the new key can be done securely using these two lower level keys. Only the members who have to know this key can decode these messages because no one except them have these keys.

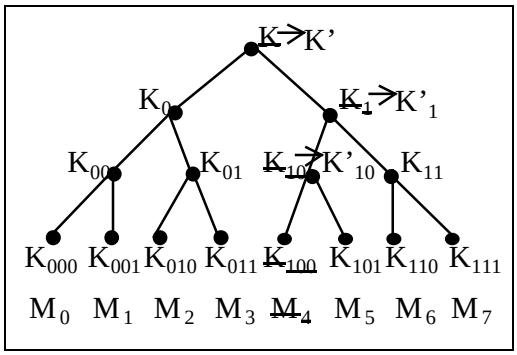

Figure 3. When member M_4 leaves, all the keys in KS_4 have to be changed (or cancelled: K_{100}).

Algorithm 2 ***Leave***(M_i);

/* The GC changes the keys in KS_i and */
/* broadcast them to the group members */

/* select a new key */
$K_{ml\cdots m1} = K'_{ml\cdots m1}$;

/* send it to M_i neighbour */
GC_broadcast($E_{K_{ml\ldots\overline{m0}}}(K_{m_l\ldots m_1})$);

for j = 2 **to** *l* **do**

/* select a new key */
$K_{ml\cdots mj} = K'_{ml\cdots mj}$

/* broadcast to half sub-tree */
GC_broadcast($E_{K_{ml\ldots m_{(j+1)}}}(K_{m_l\ldots mj})$);
/* broadcast to the other half sub-tree */

GC_broadcast($E_{K_{ml\ldots\overline{m_{(j+1)}}}}(K_{m_l\ldots mj})$);

end-for;

$K = K'$; /* select a new group key */
/* broadcast to 1st half-tree */
GC_broadcast($E_{K0}(K)$);
/* broadcast to 2nd half-tree */
GC_broadcast($E_{K1}(K)$) ;

This algorithm broadcasts two messages for each layer of the tree except for the lowest layer where only one message is broadcasted (since M_i left the

group). The total number of messages is equal to $2*(l+1)-1 = 2*\log(|M|)-1$.

3.4 Re-keying

To periodically change the group key the old group key can be used to protect the transmission of the new one.

Algorithm 3 ***Re-key***();

GC_send($E_{Kg\_old}(K_{g\_new})$);
/* broadcasted to the group */

3.5 Group size increase

The proposed key distribution algorithm will start with a reasonable tree size. However, the number of joining member may exceed the number of tree-leaves. When this happens the key distribution has to accommodate this change efficiently. We provide a mechanism for doubling the size of the tree. This action is done when the current tree is full and the group controller received a request from a new member to join the group.

Increasing the size of the tree is done by adding a new layer on top of the current tree. This allows to have another sub-tree of size equal to the previous one. The group members identifiers will be coded over $l+1$ bits. All to previous keys are renamed from $K_{ml\ldots mj}$ to $K_{0ml\ldots mj}$ (the keys indices are prefixed by a 0). The new joining members will be located on the right sub-tree, will have identifiers starting with 1 ($M_{1ml\ldots m0}$), and keys starting with one ($K_{1ml\ldots mj}$). The old group key K is renamed K_0 and a new group key K is generated.

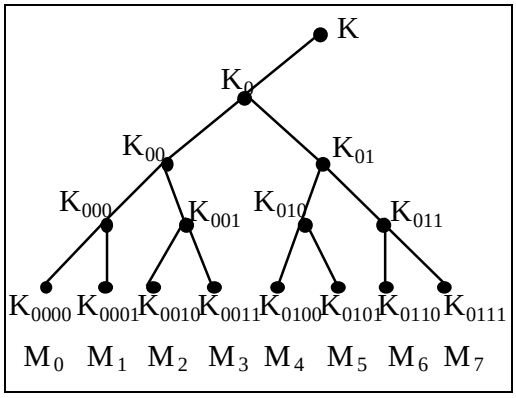

Figure 4. The group size can be doubled by adding one layer in the tree. The keys are renamed to correspond to the new numbering. This is done by prefixing all the previous group keys by a 0. The zero indicates that they are used in the left sub-tree. A new group key K is introduced.

The group size increase is executed only when the actual group members cardinal is equal to the maximum number of leaves in the tree. When a member leaves the group its leaf may be reallocated to the next member joining the group.

4 Conclusion

We have shown that TCP connections optimized for satellite links can be secured using IP. We also provided recommendations for optimizing TCP without interfering with IPsec. We also introduced a scalable key distribution algorithm that can handle large dynamic multicast groups.

References

[HC98] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)", draft-ietf-ipsec-isakmp-oakley-06.txt, February 1998.

[KA97] Stephen Kent, Randall Atkinson, "Security Architecture for the Internet Protocol", draft-ietf-ipsec-arch-sec-02.txt, November 1997.

[Mitt97] Suvo Mittra, "Iolus: A Framework for Scalable Secure Multicasting", In Proceedings of the ACM SIGCOMM'97.

[MTU98] H. Maruyama, T. Tokuyama, N. Uramoto, "A Key Update Method for Secure Multiparty Communication", Proceedings of the IEEE Globecom Internet Mini-Conference, Sydney, November 1998.

[Hurl98] P. Hurley, "Optimizing TCP over Satellite Links", European Space Agency Project, Work package 10 report, May 1998.

[Noub98] G. Noubir, "Optimizing Multicast Security over Satellite Links", European Space Agency Project, Work package 20 report, version 0.1, April 1998.

[Noub99] G. Noubir, "A Scalable Key Distribution Scheme for Large Dynamic Multicast Groups", In Proceedings of ERSADS, January 1999, Madeira island.