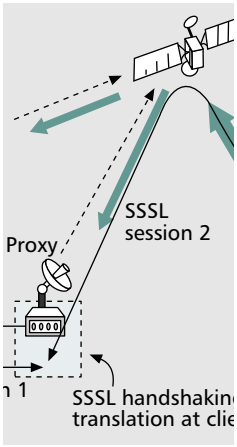
ACCEPTED FROM OPEN CALL

SECURITY ISSUES IN HYBRID NETWORKS WITH A SATELLITE COMPONENT

AYAN ROY-CHOWDHURY, JOHN S. BARAS, MICHAEL HADJITHEODOSIOU, AND SPYRO PAPADEMETRIOU, UNIVERSITY OF MARYLAND AT COLLEGE PARK



Most future networks will be hybrid in nature — having terrestrial nodes interconnected by satellite links. Security is an important concern in such networks, since the satellite segment is susceptible to a host of attacks, including eavesdropping, session hijacking and data corruption.

ABSTRACT

Satellites are expected to play an increasingly important role in providing broadband Internet services over long distances in an efficient manner. Most future networks will be hybrid in nature — having terrestrial nodes interconnected by satellite links. Security is an important concern in such networks, since the satellite segment is susceptible to a host of attacks, including eavesdropping, session hijacking and data corruption. In this article we address the issue of securing communication in satellite networks. We discuss various security attacks that are possible in hybrid satellite networks, and survey the different solutions proposed to secure data communications in these networks. We look at the performance problems arising in hybrid networks due to security additions like Internet Security Protocol (IPSec) or Secure Socket Layer (SSL), and suggest solutions to performance-related problems. We also point out important drawbacks in the proposed solutions, and suggest a hierarchical key-management approach for adding data security to group communication in hybrid networks.

INTRODUCTION

With the rapid growth of the Internet, satellite networks are increasingly being used to deliver Internet services to large numbers of geographically dispersed users. The primary advantage of satellite networks is their wide broadcast reach — a satellite can reach users in remote areas where terrestrial connectivity is not available. Satellite networks are also easily and quickly deployed, and can be a more cost-effective solution in areas where laying ground fiber networks would be too expensive.

Although satellite networks offer great potential, they also present significant challenges that need to be addressed. Security is becoming an increasingly important aspect of all network. In this article we focus on the challenges that need to be addressed in order to make satellite networks more secure while maintaining seamless interoperability with terrestrial networks. These security-related challenges include the following considerations:

- Satellite channels are wireless broadcast media, which makes it possible for an unauthorized user to receive the signal and eavesdrop on the communication, if it is not encrypted.
- Without proper security mechanisms, any sufficiently well-equipped adversary can send spurious commands to the satellite and jam or disrupt the communication.
- Satellite channels can occasionally have high bursty errors (for example, during heavy rain) that result in packet loss. Satellite networks also suffer from long propagation delays (for example, 0.5 seconds for geostationary satellites). Therefore, security systems should add minimal delays to the communication and have mechanisms to recover from loss in security information.

Incorporating security solutions originally designed for terrestrial networks, such as Internet Security Protocol (IPSec) or Secure Socket Layer (SSL), into satellite networks can cause severe performance penalties.

In this article we consider some of these issues. We focus on data security for IP-based commercial networks, and discuss the performance problems that arise due to the encryption of the Transmission Control Protocol (TCP) header and payload when popular unicast security protocols like IPSec or SSL, originally designed for terrestrial connections, are applied to satellite networks without incorporating changes necessitated by the unique characteristics of satellite networks. We also look at the protocols proposed for secure group communication in hybrid satellite networks, and describe a hierarchical approach to group key management that is robust, scalable, and suitable for the characteristic topology of hybrid networks.

The rest of the article is organized as follows. We describe the hybrid satellite-network topology and features that make it different from terrestrial networks. We discuss security needs for the hybrid network. We discuss the current approach to provide end-to-end unicast security in hybrid networks, and describe the performance problems arising as a result. We survey

1536-1284/05/$20.00 © 2005 IEEE

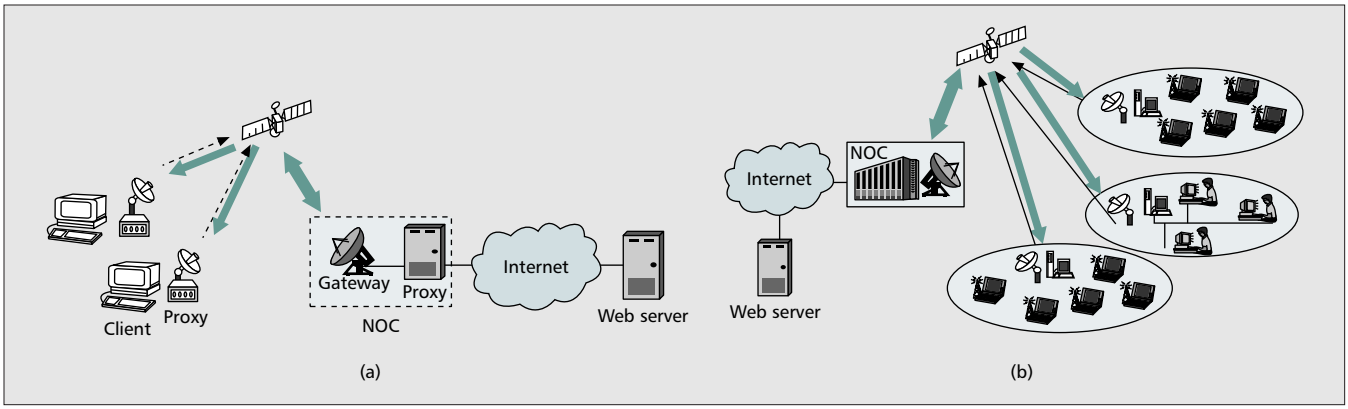


Figure 1. *Commercial direct-to-home network topology: a) case 1; b) case 2.*

the proposals for key management for secure group communication in satellite networks. We describe a possible solution to secure unicast communication without sacrificing performance and highlight our key-management approach to security for group communication in satellite networks. We conclude the article by pointing to future research directions.

Commercial Hybrid Satellite Network Architecture

The network topologies we consider are illustrated in Fig. 1. In both topologies, we assume that there is one geostationary satellite with multiple spot-beams covering a large geographical area. Each spot-beam covers a subset of the total user set. We assume that future satellites will have an IP stack, be capable of onboard processing, and switch the data between supported spotbeams. The satellite therefore acts as an *IP router-in-the-sky*. The Network Operations/Control Center (commonly known as NOC or NCC) connects to the satellite through the hub satellite gateway. The NOC is also connected to the Internet through high-speed terrestrial links.

Terrestrial users can be either standalone machines (Fig. 1a), or a cluster of machines at each location, such as a local area network (LAN) (Fig. 1b). Terrestrial LANs can be either wired or wireless. Each user or LAN is connected to a local satellite terminal. The users receive traffic from the satellite via the forward channel (satellite downlink). The users can also communicate with the satellite via the return channel (uplink). There is no terrestrial connectivity between the users or the LANs.

Usually, in commercial satellite networks that transfer Internet traffic, a split-connection Transmission Control Protocol (TCP) Performance Enhancing Proxy (PEP) is implemented to reduce the negative effects of the satellite link on the Internet connection [1]. Satellite channels provide large bandwidth (which can be as high as 90 Mb/s in the downlink), but also suffer from long propagation delay in comparison to terrestrial links. The delay can be as high as 500 ms (round-trip) for a geostationary satellite link. The propagation delay can have a severe adverse impact on the delivery of Internet traffic. Most of the Internet traffic uses the TCP, which is highly susceptible to the delay-bandwidth product and exhibits very poor performance in satellite channels. Satellite TCP connections need large transmit windows to fully utilize the available bandwidth. However, due to the TCP *slow-start algorithm* and large propagation delay in the satellite channel, it takes much longer for satellite TCP connections to reach the target window size, in comparison to terrestrial TCP connections. Also, the window is very vulnerable to congestion due to the multiplicative decrease strategy of TCP. The problem is compounded by the fact that TCP misinterprets link-layer corruption (which is the prevalent source of loss in satellite links) as congestion (which is rare) and consequently reduces the window.

The PEP provides an efficient solution to the above problem. In satellite networks, a PEP agent is installed at the satellite gateway between the satellite network and the Internet. The PEP agent inspects every TCP packet that flows through the network. For data packets, the PEP sends back premature acknowledgments to the TCP senders, without waiting for the TCP segments to be actually delivered to the receivers. These premature acknowledgments are specially formatted to be indistinguishable from real acknowledgments and they considerably shorten the perceived round-trip delay. Studies have shown that this technique is critical for the performance improvement of satellite networks [2–4]. Hence, TCP PEPs have been widely deployed in satellite networks today.

Commercial networks also employ HTTP proxy servers to improve the speed of responses to Web-browser requests. When a user browses through content on the Internet, the application layer protocol in use is HTTP. A typical HTTP exchange involves a request by the browser for a Web page ("GET"), and a response from the Web server, which contains the hypertext markup language (HTML) text of the requested Web page. A typical HTML page would also contain multiple embedded "objects" such as images, embedded media or scripts, and so forth. Each embedded object has to be retrieved with a separate HTTP request-and-response exchange. Therefore, a Web page that contains $n - 1$ embedded objects takes $n * RTT$ time to load fully, where RTT is one round-trip time. This can be extremely costly in a satellite network, where the RTT is usually high.

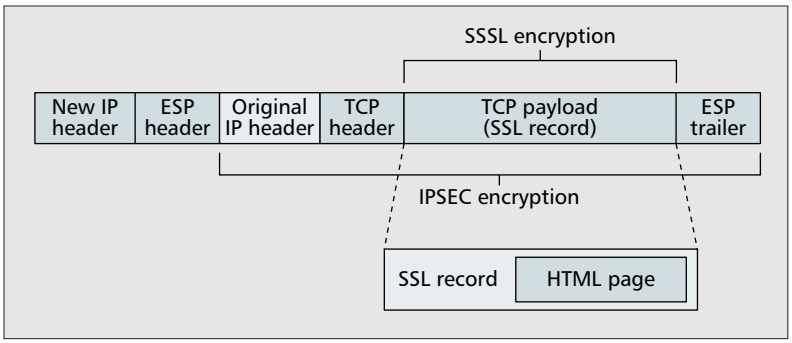


Figure 2. *IPSec and SSL encryption on a packet.*

The HTTP proxy server (also known by various other names, depending on the vendor) is implemented in satellite networks to overcome this problem. In a typical implementation, this requires a local Web proxy server at each user location, and a remote proxy server at the central hub facility of the satellite network (i.e., the NOC). The Web browser at the user location should be able to recognize the local proxy (which can be either software on the client machine, or a separate hardware connected in-between the client machine and the local satellite terminal). When the browser makes a request for a Web page, the HTTP GET request is sent to the local Web proxy, which forwards the request to the destination Web server. The Web server responds with the requested base HTML page. This page is intercepted by the proxy server at the network hub facility. The hub proxy server reads the base HTML page and sends multiple GET requests to the destination Web server for all the embedded objects in the base HTML page. This exchange occurs over a high-speed terrestrial connection between the hub and the Internet, thereby saving the time each request would have needed for a round trip over the satellite link. As the objects of the Web page are retrieved by the hub, they are immediately forwarded to the proxy at the user location. As the user browser receives the base HTML documents, it generates appropriate GET requests to fetch the objects corresponding to the links embedded in the document. The browser GET requests are terminated at the Web proxy server, which forwards the prefetched documents to the user browser immediately. The net result is that only a single "GET" request from the user browser traverses the satellite link, while a set of rapid responses quickly deliver the requested Web page and associated elements to the browser. The need for satellite capacity is also reduced, which is the most costly element of a satellite network. In terms of the user's experience, the user sees a brief pause after the original Web-page request (corresponding to the round-trip time it takes for the request to the forwarded to the destination server, and the response to be received by the browser, over the satellite link), followed by near-instantaneous delivery of all content residing on the requested page. The trade-off is additional hardware at the user location and the central-hub facility.

In Fig. 1a, the proxy server at the user represents both the PEP (user side) and the HTTP proxy (user side). There is a hub proxy server located at the NOC with the hub satellite gateway — this proxy server represents the gateway proxy for both TCP and HTTP performance enhancements.

Security Threats

Similar security attacks can be launched against different hybrid satellite network topologies, but the impact of attacks would differ depending on the type of network and the applications supported by the network scenario. In the following, we list some of the important security threats in the hybrid network described above, and highlight the importance of the threats for the different network scenarios.

Confidentiality of information: For networks that require information privacy, a primary threat is unauthorized access to confidential data or eavesdropping. Since the satellite is a broadcast medium, any entity on the ground with the right equipment can receive the satellite transmission. If the data is broadcast in the clear, then adversaries can be privy to the information that is flowing in the network.

Data confidentiality can be achieved by message encryption. This requires that the senders and receivers are concurrently aware of the correct cryptographic keys used in the encryption/decryption operations. This is a twofold problem: the problem of selecting suitable cryptographic algorithms for doing encryption so that overall network performance is not affected, and the problem of coordinating keys between users, that is, *key management*.

Sending spurious commands: An adversary with the right equipment can send spurious control and command messages to the spacecraft, thus making the spacecraft perform operations different from their intended use. This can disrupt legitimate operations and communication in the network.

This attack can be prevented if the sources of the messages are properly authenticated by every receiver. This would require suitable mechanisms for authentication, such as digital signatures [5]. The level of security required would dictate the authentication policy, for example, whether only the end users should authenticate each other, or whether authentication should happen on a per-hop basis. The latter might be necessary for scenarios where the satellite should not broadcast spurious information. If the satellite authenticates the source of every message it receives, it will transmit only those messages for which source authentication occurs correctly.

Message modification attack: When the traffic goes over open networks, an adversary who is listening on the path can intercept both control and data messages. The adversary can modify the messages and send them to the destination, which can be the spacecraft, the ground terminals, or the end users. When the message reaches the intended destination, it would think that the corrupt message is coming from the true source, but the message content might be different from that expected or required for normal network operation.

Message modification can be prevented by

appending message-integrity check mechanisms to every message, for example, message authentication codes (MACs) [6] or digital signatures. Security requirements and policies can dictate whether message authentication should happen only at the communication end points, or whether intermediate nodes should also verify the integrity of every message.

Denial-of-service attack: Some attacks on security can be facilitated if strong security mechanisms are put in place for performing message-integrity checks or authenticating users. Consider the case where the satellite does authentication and integrity checks on all messages before broadcasting. An adversary can send a large number of spurious messages to the satellite, thus making the satellite spend significant computational cycles processing the spurious messages, which could be better spent broadcasting legitimate messages. Since the satellite has limited processing power, such an attack can be very effective, especially if strong cryptographic mechanisms like digital signatures are used for authentication and message integrity. This is a denial-of-service (DOS) attack. Although this DOS attack can be launched against any node in a network, a satellite network can be particularly susceptible to such an attack, since the satellite is a single point of failure and can be easily overwhelmed if made to perform too much computation.

Securing End-to-End Unicast Communication Using IPSec or SSL

Research on satellite security has focused on using the existing standardized technology, originally designed for terrestrial networks, to fix well-known security holes in satellite networks. Two such protocols that are widely used for secure unicast communication are IPSec [7] and SSL [8]. Figure 2 illustrates the encryption regions of SSL and IPSec.

Secure Socket Layer for Secure Web Traffic

The SSL protocol secures the Web-browsing connection on an as-needed basis. When the client requests a secure connection or the server demands one, SSL is activated to secure the HTTP connection. The resulting connection is popularly known as *secure HTTP* (or HTTPS) and it encrypts the application-layer HTTP data end-to-end between the client and the server. In the protocol stack, the SSL layer sits between the application and the transport layers. Therefore, SSL encryption hides the TCP payload from all nodes in the network, except the client and the server.

SSL encryption does not allow the HTTP proxy to function correctly. The HTML Web page encrypted into the SSL records is readable only by the client and the server who have the decryption keys. The keys are not available to the proxy, and therefore the proxy cannot read the HTML Web page. Consequently, the hub proxy server cannot send requests to the Web server for the embedded objects in the page and, therefore, HTML object prefetching cannot take place. The net result is that a Web page with n – 1 embedded objects takes $n * RTT$ to be loaded, an increase in delay by a factor of n.

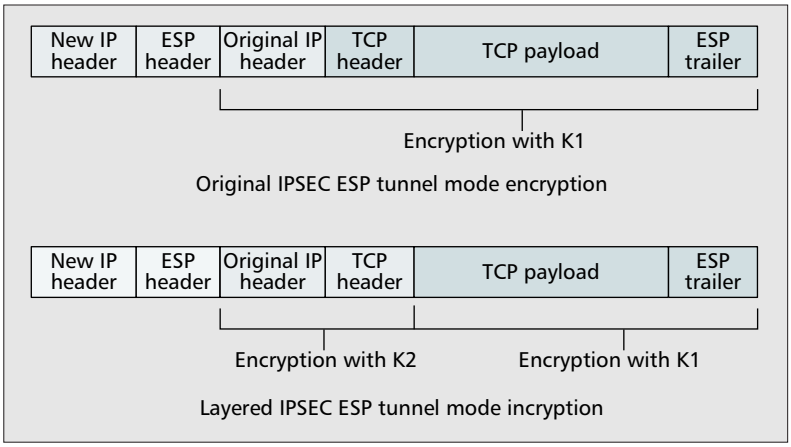


Figure 3. *IPSec and layered IPSec encryption. Key K1 is shared between endpoints only. Key K2 is shared between endpoints and TCP PEPs.*

IPSec for Security at the Network Layer

Several proposals for data confidentiality and authentication in satellite networks call for use of IPSec, which has been widely adopted by the Internet Engineering Task Force (IETF) for security at the network layer. IPSec and SSL are used independently of each other. IPSec creates an end-to-end *tunnel* at the network layer for the secure transfer of traffic. The two end-points in the communication negotiate security parameters known as the security association (SA) before traffic can be encrypted. Once the SA has been established in the handshake phase, the IP packets are encrypted using the algorithms and the keys specified in the SA. This is done when the IP-encrypted security payload (IPSec ESP) [9] is used. The IPSec ESP provides for both data encryption and authentication.

IPSec provides strong security for data confidentiality and authentication, but it has a heavy byte overhead — in the ESP mode, IPSec adds 10 bytes of overhead to the header and trailer. In addition, if authentication is used, ESP adds 16 bytes or more for the integrity check value, and another 8 bytes or more of initialization vector (IV) if the encryption algorithm uses an IV. Also, IPSec has been designed primarily to secure point-to-point communication; it is not well suited for group communication, due to the lack of the dynamic key-establishment procedure necessary to for secure communication in groups where the membership changes with time. In addition, IPSec does not allow for authentication at intermediate nodes, but this might be useful in some security situations.

A widely researched problem when using IPSec in satellite networks is its inability to coexist with PEPs. The keys used for encryption in the IPSec ESP are known only to the two endpoints and therefore any intermediate node in the network cannot decrypt the traffic. IPSec ESP has two modes of operation — tunnel mode and *transport* mode. In tunnel mode, the entire IP packet is encrypted and a new IP header and ESP header are generated and attached to the encrypted packet (Fig. 3), which adds an extra

The HTTP proxy also cannot function when the IPSec ESP is used. Since the HTML page is encrypted end-to-end, the HTTP proxy cannot read the Web page in order to prefetch the embedded objects. Therefore, use of IPSec leads to a severe degradation in performance for both the TCP PEP and HTTP proxy.

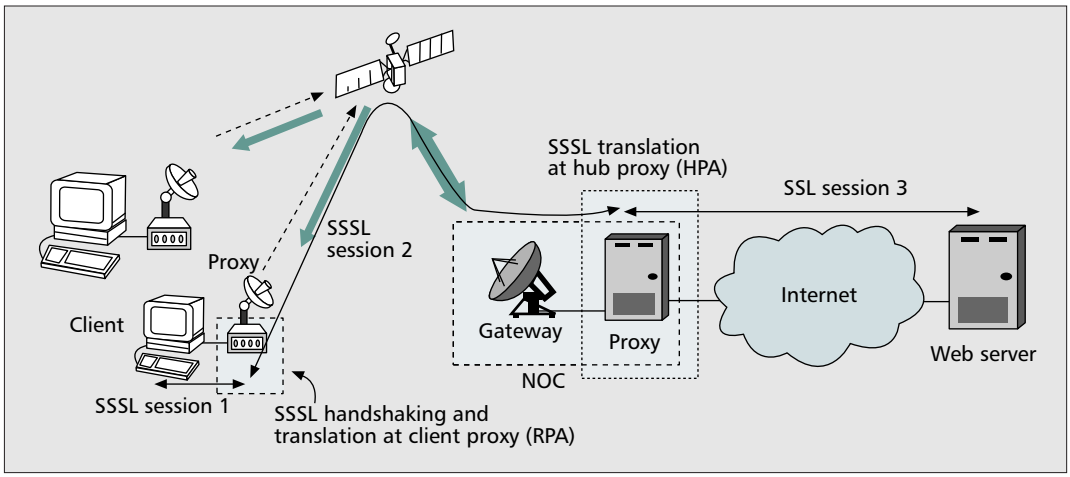


Figure 4. *The SSL Internet Page Accelerator concept for efficient HTTPS over satellite.*

20 bytes of overhead in addition to the overhead mentioned above. Encrypting the original IP header provides very strong security by disabling attacks (such as traffic analysis, etc.). In transport mode, the payload portion of the IP packet is encrypted and a new ESP header is attached to the packet after the original IP header, which is in the clear. In either mode, the IP packet payload, which includes the TCP header, is encrypted with keys known only to the end points. Therefore, a TCP PEP, which is an intermediate node in the communication path, cannot read or modify the TCP header, since the PEP does not know the keys. Consequently, the PEP cannot function, thus leading to degradation in the performance of the TCP protocol.

The HTTP proxy also cannot function when the IPSec ESP is used. Since the HTML page is encrypted end-to-end, the HTTP proxy cannot read the Web page in order to prefetch the embedded objects. Therefore, use of IPSec leads to a severe degradation in performance for both the TCP PEP and HTTP proxy.

It is important to note that the problems that arise from the use of the SSL protocol or the IPSec ESP are independent of one another. It is conceivable that both protocols are used simultaneously, for example, when a secure Web page is accessed via a secure VPN tunnel. However, in such cases the performance issues do not change and the effect would be equivalent to using the IPSec ESP alone. On the other hand, if SSL alone is used, then the performance would be better, since the TCP PEP can function correctly in this scenario.

Proposed Solutions to Mitigate Performance Problems with SSL or IPSec

Several proposals have been made in academia and industry to deal with performance problems that arise from using IPSec and SSL in satellite networks.

The concept of breaking up IPSec encryption into multiple encryption regions or zones on a single packet has been proposed independently in [10, 11]. Although the finer details in the two approaches are different, the basic idea is the same. Known as multilayer IPSec (ML-IPSec) [10] and layered IPSec [11], the idea is to encrypt different regions of the IP packet using different keys (Fig. 3). The TCP payload is encrypted with key $K1$, which is shared only between the endpoints. The original IP header and the TCP header are encrypted with key $K2$, which is shared between the end points and also with intermediate authorized nodes such as the TCP PEP. Therefore, the TCP PEP can decrypt the header portion of the ESP packet with $K2$ and read the TCP header to do its performance optimizations. But the PEP cannot read the TCP payload and thus cannot access the actual data, since it does not posses the key $K1$.

The layered IPSec approach allows TCP PEPs to function effectively. However, the method does not solve the problem of HTTP proxy servers. The HTML page is encrypted with key $K1$ as part of the TCP payload, and $K1$ is not shared with any intermediate node. Therefore, the Web page is not accessible to the HTTP proxy and no object prefetching can be accomplished.

Olechna *et al.* [12] have suggested two solutions to the IPSec problem. In the first approach, the paper proposes moving the TCP PEP gateways to the endpoints. The TCP optimizations are done on the traffic in the clear, and then the traffic is encrypted using IPSec. There is no TCP PEP at the satellite hub. This approach improves the performance, but when a packet is lost or received in error TCP goes into congestion-avoidance phase and the transmission is reduced by half. The second proposed approach, which deals effectively with this problem, is to split the secure connection into two at the satellite gateway. One connection is between the client and the gateway, and the second connection is between the gateway and the Internet server. This allows the gateway to decrypt the IPSec packet and read the headers and thereby do performance optimizations. This requires trust in the satellite gateway, which can now read all the traffic. This might be unacceptable to users who require strong end-to-end security.

Several modified TCP protocols have been proposed that perform better than the original specification in the event of channel errors or delay, or when IPSec is used. A discussion of

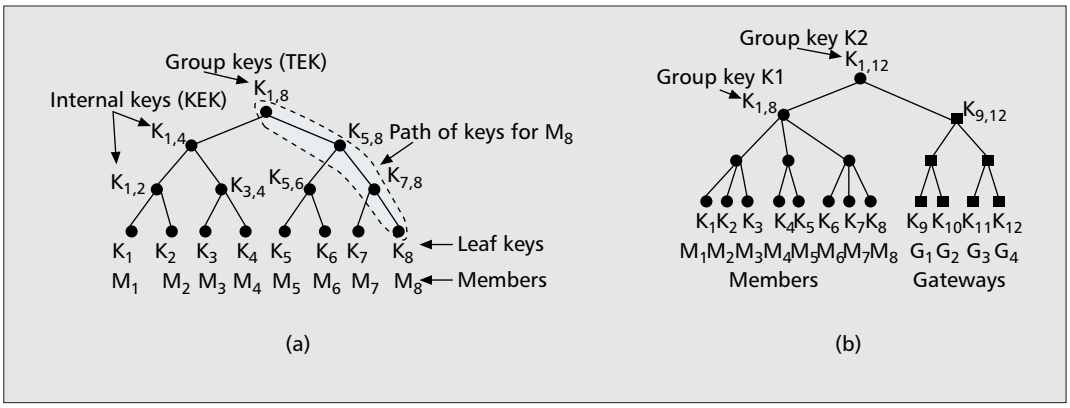


Figure 5. *Logical key hierarchy and its extension to satellite networks: a) with eight members; b) ML-IPSec integrated LKH tree with users and gateways.*

these TCP enhancements can be found in [13].

The problem of HTTP proxy performance when SSL is used has been addressed within the industry by breaking up the end-to-end single SSL connection between client and server into multiple SSL connections [14]. In this solution, the client browser creates a secure HTTP connection with the remote page accelerator (RPA) at the client satellite terminal, a second connection is created between the RPA and the hub page accelerator (HPA), and a third connection is between the HPA and the server (Fig. 4). The RPA performs all necessary handshaking with the client browser. The HPA can decrypt the SSL traffic from the server and perform the desired object prefetching. Taken together, this allows delivery of secure Web content with little performance degradation and with little change to the standard protocols. The major drawback to this scheme is that it requires a high level of trust in the intermediate nodes. The HPA, which is a third-party entity, can read all the sensitive Web traffic that passes between the client and the server. This might be unacceptable when absolute end-to-end security is desired.

Key Management Proposals for Secure Group Communication in Hybrid Networks

Some research has been done with individual algorithms that serve as tools in building key-management protocols in order to facilitate secure group communication in hybrid satellite networks.

Howarth *et al.* [15] have proposed the use of logical key hierarchy (LKH) [16, 17] for efficient key management for multicast groups in a satellite network. LKH makes use of a centralized key manager or group controller (GC), which constructs a logical key tree with the group members as the leaves of the tree (Fig. 5a). The internal nodes of the tree are the key encrypting keys (KEK), which are used to securely transport key updates to the group. The root of the tree is the session key or traffic-encrypting key (TEK), which is used to encrypt the session traffic. The number of keys that need to be updated when a member node joins or leaves the group is $O(logN)$ (where N is the number of members in the group), which is less than the $O(N)$ keys required if the GC arranged the members in a flat topology.

To allow PEPs to function correctly when network-layer security is used, [15] proposes the use of ML-IPSec. The paper proposes using a single LKH tree to manage the group key $K2$, used to encrypt the transport layer header (known to end users and trusted gateways), and the group key $K1$, known only to the end users and used for encrypting the transport layer data. As shown in Fig. 5b, users $M_1 \ldots M_8$ are leaf nodes in a subtree of degree three, and gateways $G_1 \ldots G_4$ are leaf nodes in a subtree of degree two. The root key of the member node subtree, $K_{1,8}$, is used to encrypt the transport payload. The root of the overall key tree, $K_{1,12}$, is used to encrypt the transport header. All member nodes know both $K_{1,8}$ and $K_{1,12}$, but the gateways know $K_{1,12}$ only (apart from the internal keys in the gateway subtree).

How the LKH tree would be managed is not stated in [15]. This is important, since the users and the gateways might not be in the same administrative or security domain. The paper also considers all users and gateways as a "flat" network for key distribution purposes, rather than taking into account the hierarchical nature of the network topology.

The use of LKH for key management in satellite links has also been proposed in [18], which suggests algorithms for dynamically managing the LKH tree in case of member joins and leaves.

Duquerroy *et al.* [19] proposed "SatIPSec," for key distribution and secure communication for both unicast and multicast in a satellite network. The solution is based on IPSec, with the addition of flat multicast key exchange (FMKE) to support key management for secure group communication. Management of SAs for both unicast and multicast communication is integrated into the FMKE protocol. FMKE also incorporates reliability mechanisms so as to guarantee reliable key distribution in the lossy satellite setting. However, FMKE manages SAs between the satellite terminals or gateways only and does not extend to the end users. Therefore, end-to-end security is not provided when using SatIPSec.

The RPA performs all necessary handshaking with the client browser. The HPA can decrypt the SSL traffic from the server and perform the desired object prefetching. Taken together, this allows delivery of secure Web content with little performance degradation and with little change to the standard protocols.

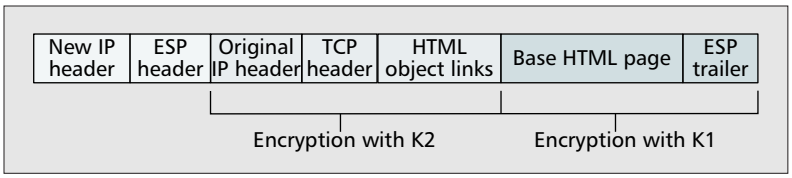


Figure 6. *Layered IPSec with modifications for HTTP optimization.*

Also, FMKE treats all the satellite terminals it services (which are called SatIPSec clients) in a "flat" topology, and establishes separate secure channels to all SatIPSec clients. This will not scale when there are a large number of clients. Also, SatIPSec does not consider the dynamic joins and leaves of members in the group communication setting; a client needs to be preauthorized for all the groups it wants to take part in. The protocol also requires complete trust in the group controller and key server (GCKS), which is a third party that is responsible for managing the SAs between the clients. All clients need to have preshared secrets with the GCKS.

IPSec and SSL in Hybrid Networks: Our Approach

We look at separate solutions to the performance problem arising out of using SSL and IPSec in hybrid networks, and also consider how the two approaches can be combined.

HTTP over IPSec Tunnel

One viable method is to break up the end-to-end IPSec tunnel into multiple connections. This is similar to the solution proposed in [12]. But while their approach looks at only the TCP enhancements, we add the use of the HTTP proxy as well. In our approach, the IPSec connection from the client is terminated at the client proxy. The proxy creates its own IPSec connection to the gateway TCP proxy. A third IPSec connection is created from the gateway TCP proxy to the Web server. Schematically, this is similar to Fig. 4, with IPSec connections replacing the SSL connections in the figure. The IPSec handshaking between the client and the server is spoofed by the client proxy on the client end, and by the TCP hub proxy on the server end. In this model, the Web traffic can be read completely by the client proxy and the hub proxy. The two proxies are able to perform the TCP enhancements because they can read the TCP header. In addition, the hub HTTP proxy can perform HTML object prefetching from the server because it can read the base HTML page as it is returned to the client on a HTTP request. When the client browser generates staggered requests for the embedded objects upon receiving the base HTML page, the client proxy is responsible for returning local acknowledgments to the requests, and sending all the objects to the client browser at one time. The design is therefore fully able to maintain the functionality of the TCP and HTTP proxies. It also encrypts the traffic so that it can be seen only by the client, the server, and the two intermediate proxy servers. The design also makes minimal changes to existing standard protocols. However, the design also requires that there be full trust in the proxy servers. Also, there is additional overhead in setting up three IPSec connections, as opposed to one (as in the end-to-end case). The overhead in encryption/decryption also increases by a factor of three for every IP packet, since the intermediate proxies need to decrypt the TCP header and the HTML content.

When the security requirement is that the traffic be unreadable to intermediate nodes, the above approach will not work. In this situation, we propose extending the layered IPSec approach in order to allow portions of the HTML content to be also accessible to the proxy servers. Assume for layered IPSec that the keys are $K1$ and $K2$. $K1$ is known only to the client and the server, while $K2$ is known to the client, the Web server, and the intermediate proxy servers at the client and the gateway. When the client makes HTTP requests, the requests are encrypted using $K2$, so that the client proxy server can read the requests and send local acknowledgments. Additional software at the Web server parses the requested HTML page so as to obtain all the embedded object links. These object links are collated into a new HTML page that contains only the object links, and this new page is encrypted with $K2$. The base HTML page that contains all the information and the object links is encrypted with $K1$. Both the encrypted base HTML page and the encrypted object links HTML page are sent in reply. Therefore, the encrypted ESP packet looks as it is depicted in Fig. 6. Upon receiving the IPSec packet from the Web server, the hub proxy is able to read the object links (since it has $K2$) and therefore do prefetching for the embedded links. In addition, the hub proxy can also read the TCP header and perform TCP enhancements. However, the HTML base-page data cannot be read by the hub proxy, since it does not have $K1$. The encrypted base HTML page can only be read by the client when the IPSec packet reaches the destination.

This design allows the TCP and HTTP proxies to perform effectively while maintaining a high level of end-to-end security. However, the security is not as strong as in traditional IPSec, since the intermediate proxies do get some information insofar as they can read the links of the embedded objects, even though they cannot read the application data. This is the major trade-off necessary to achieve acceptable performance in this design. In addition, the model requires changes to be made to the IPSec protocol so that layered IPSec is supported with the HTTP performance additions.

A major issue in the above model is the handshaking mechanism required to set up the layered IPSec connection. To maintain a high level of security, we propose that the connection be set up primarily between the client and the server, who negotiate both $K1$ and $K2$, apart from other parameters of the security association. The handshaking mechanism then provides $K2$ securely to both the client and the hub proxy servers. The client and the hub proxy servers are required to authenticate themselves correctly before they can receive the secondary key or access the IPSec traffic.

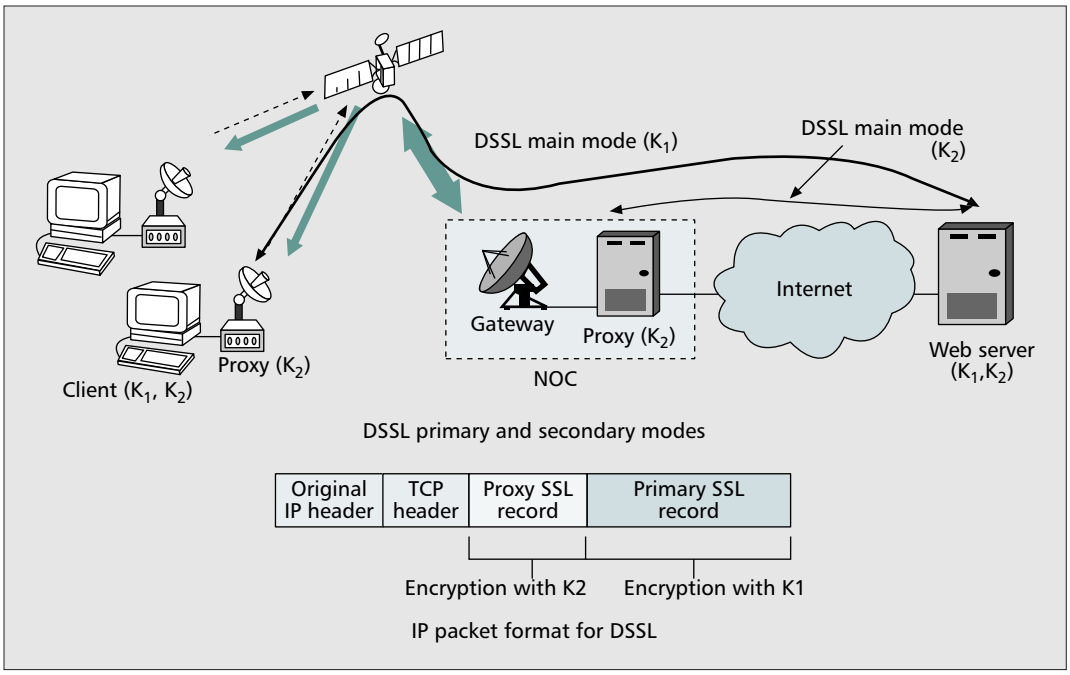


Figure 7. *Dual-mode SSL for HTTP optimization.*

The DSSL design is more complex in comparison to SSL since it requires the creation of an additional connection, and therefore involves a higher overhead. There is also the added overhead of multiple encryptions and decryptions with two different keys.

HTTP OVER SSL

When the HTTP traffic is secured using SSL only, and there is no IPSec tunnel in use, several approaches are possible to ensure acceptable performance. If the security requirement of the client and the Web server allow for trusted intermediate nodes, then the SSL accelerator concept of [14] can be a viable solution. This would require no change to the protocols at the expense of higher overhead in order to set up multiple SSL connections between the client, proxy, and Web server.

When the security policy does not allow for trusted third parties, a different approach is needed. We propose the use of a modified SSL protocol, which we term dual-mode SSL (DSSL). As shown in Fig. 7, the secure connection in DSSL has two modes — an end-to-end *main* mode connection between the client and the Web server, and a *secondary* mode connection that has the hub HTTP proxy as an intermediate node. When secure HTTP traffic is requested, the DSSL main mode connection is first negotiated between the client and the server. As part of the handshake for the main mode, the client and the Web server also negotiate the parameters for the secondary mode. Let $K1$ be the encryption key for the main mode, and $K2$ be the encryption key for the secondary mode. The client transfers the parameters of the secondary mode to the client and hub HTTP proxy servers only after the proxy servers authenticate themselves to the client. When the client makes an HTTP request, the client proxy sends local replies to the client browser, as discussed previously. The Web server, on receiving the request, parses the requested HTML page to obtain the embedded object links, which are collated into a new HTML page. The object links HTML page is then encrypted by DSSL using $K2$ to create the proxy SSL record. DSSL encrypts the base HTML page using $K1$ to create the primary SSL record. The two records are appended together and sent to the client in an IP packet (Fig. 7). The hub proxy intercepts the IP packet, extracts the object links from the proxy SSL record using $K2$, and prefetches the embedded objects. The Web server always encrypts the actual objects using $K1$, so that the hub proxy cannot read the base HTML page data. The hub proxy transfers all the embedded objects together to the client at one time. Therefore, the HTTP proxy functionality is preserved in DSSL while maintaining the end-to-end security of the HTML page contents. However, the security is less than in the end-to-end SSL connection case, since the HTTP proxy can read the object links. In standard SSL, the proxy servers can read no part of the base HTML page, not even the object links. We believe this slight reduction in security is acceptable, given the considerable improvement in performance using this method.

The DSSL design is more complex in comparison to SSL since it requires the creation of an additional connection, and therefore involves a higher overhead. There is also the added overhead of multiple encryptions and decryptions with two different keys, and the complexity of parsing the HTML page for the object links. All these require changes to the base SSL protocol.

The DSSL concept is similar to the multiple-channel SSL concept proposed in [20]. However, the authors do not differentiate encryption in primary and secondary SSL records but instead suggest that HTTP traffic with lower security requirements be encrypted entirely with keys known to intermediate nodes. For our security requirements, that approach would not be acceptable.

Differential Encryption in Single SSL Record — The use of a proxy SSL record is not necessary if various parts of the HTML page can be encrypted with

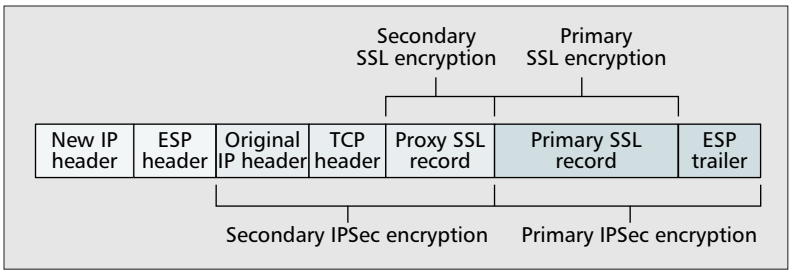


Figure 8. *Packet format for dual-mode SSL with IPSec.*

different keys. In that case, the Web server can encrypt the object links in the HTML page with key $K2$ and the rest of the HTML page contents with key $K1$, thus creating a single SSL record with different encryption. The hub proxy server can parse the SSL record and decrypt only the object links with key $K2$, before forwarding the IP packet to the client proxy. We assume that the primary and secondary encryption keys $K1$ and $K2$ have been set up and distributed as described in the previous sections, with $K1$ known to the client and the Web server only, while $K2$ is known to the client, the Web server, and the intermediate proxy servers.

A similar technique can be applied when IPSec encryption is used instead of SSL encryption. The advantage here is that the size of the packet does not increase, although there is the overhead of distributing key $K2$ to the proxy servers to be considered.

HTTPS over IPSec

For the sake of completeness, we consider the situation where a secure Web page is requested over an IPSec tunnel. This method involves redundancy of resources, since use of SSL when IPSec is being used does not provide any substantially added security. However, our approach can take care of the performance in this scenario as well.

In this situation, we propose integrating DSSL with layered IPSec. Then the secondary keys for both the layered IPSec connection and the DSSL connection are shared with the proxy servers. The secondary key for layered IPSec is shared with both the TCP proxy and the HTTP proxy. When layered IPSec encrypts the packet, the secondary key encryption extends up to the proxy SSL record. The TCP proxy servers can therefore decrypt the TCP header of the ESP packet, and the HTTP proxy server can decrypt the proxy SSL record. Consequently, performance optimizations for both TCP and HTTP are allowed without letting the intermediate servers read the HTML page. A schematic of the IPSec packet in this setting is shown in Fig. 8.

A Hierarchical Approach to Key Management for Data Security in Hybrid Networks

In [21], we have proposed a key-management framework for distributing cryptographic keys securely and in a scalable manner to users taking part in group communication in a hybrid satellite network. The objective is to ensure data confidentiality, by encrypting the data traffic with group keys known to all the group members. The key-management framework is built on top of the multicast routing architecture. We have considered the hybrid network topology shown in Fig. 1b and designed a multicast routing architecture to allow users to communicate seamlessly between multiple terrestrial LANs (also referred to as subnetworks) [22]. Our routing design makes specific use of asynchronous transfer mode (ATM) point-to-multipoint routing [23] over the satellite links, and Protocol-Independent Multicast Sparse-Mode (PIM-SM) multicast routing [24] in terrestrial LANs. We have extended PIM-SM to allow multiple rendezvous points (RPs) in each multicast group. The satellite gateway in each LAN acts as the local RP for the LAN and creates the local multicast trees for group members within the LAN. The local multicast trees are connected together over the satellite links by using the ATM *point-to-multipoint virtual connection*, thereby creating one end-to-end multicast tree for each group, encompassing all the LANs with group members in them. The multicast routing architecture is thus adapted closely to the hierarchical network topology, and allows for building efficient multicast trees with low control and data overhead.

The design of the key-management protocol is independent of the routing algorithm, although it is based on the same underlying principle, that is, a hierarchical breakup of the network based on the topology. We divide the network into two levels — the lower level, comprised of terrestrial LANs where the users are located, and a higher level consisting of the satellite, the NOC, and the satellite gateways or RPs in each LAN, which together form an overlay (Fig. 9a) interconnecting terrestrial LANs. The RPs act as the "bridge" between the two levels.

Key management is done separately in the two levels. In each LAN we introduce a local group controller (called the "subnetwork key controller" or SKC) to manage the keys for all groups active in the LAN. The SKC is responsible for access control of all members of all groups that are active in its LAN, generating the group keys for all local groups, and updating the keys on group-member joins and leaves when a group is active. The keys managed by an SKC are entirely local to its LAN, and do not affect the key management in any other LAN in the network. The SKC uses the LKH algorithm to manage keys in its LAN, creating a logical key tree that we term the *SN Tree*. Each group active in a LAN has its own SN Tree. The leaves of the SN Tree for a group correspond to the long-term shared secrets between the SKC and the local users in the LAN who are active as sources and/or receivers in the group. The root of the SN Tree corresponds to the session key that is used for encrypting the group traffic within the LAN at any particular instant. On member joins and leaves, the session key, and all the keys on the path from the root to the leaf node corresponding to the member joining/leaving, are updated, while all other keys in the SN Tree remain unchanged.

The overlay has its own key management,

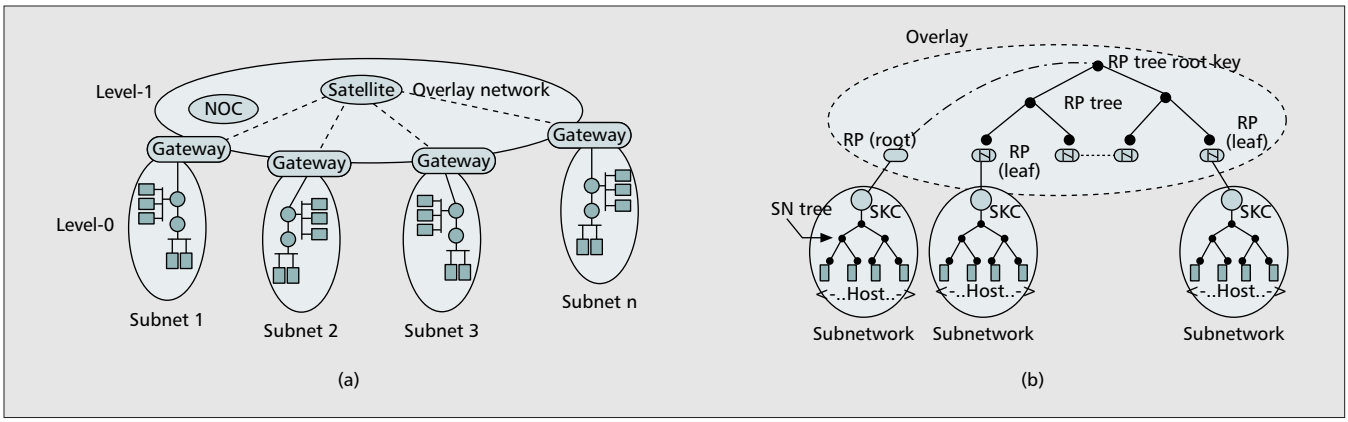


Figure 9. *A hierarchical approach to key management in hybrid networks: a) hierarchy in the hybrid network; b) tiered tree key management.*

also based on the LKH algorithm. At the overlay level, the key management for a particular group is controlled by the satellite gateway/RP (known as the *root RP* for that group) of the LAN that has group sources active for the longest continuous period in the group. The logical key tree for any group thus formed at the overlay is termed the *RP Tree*. The root RP is responsible for generating keys for the RPs of the LANs who subscribe to the particular group, that is, have sources and/or receivers active in the LAN. Each group has its own RP Tree. The design ensures that the NOC cannot receive/transmit data to any active group, unless it explicitly subscribes to the group as a member node. However, LANs joining any particular group initially register with the NOC, which maintains a *group membership table* for all active groups, so that at all times the NOC is aware of the LANs which are participating in all active groups. The NOC is also responsible for selecting the root RP of the RP Tree for each group, which it does based on the *earliest-to-join* policy. The root RP also might be different for different groups, since the LAN with the longest continuously active sources might be different for different groups. Our algorithm has the provision to allow the root RP for any group to change — this happens if the currently active root RP leaves the group, when all sources/receivers within its local LAN cease to participate in the group.

Our algorithm therefore builds a hierarchy of logical key trees that closely follow the hierarchy in the network topology, as shown in Fig. 9b. We term this framework *Tiered Tree-based Key Management*. In this hierarchy of key trees, the gateway RPs are responsible for performing *key translation* on all the multicast group traffic as it transmits the data from local sources to receivers in remote LANs, or when it receives group traffic from remote sources for local receivers. This translation is necessary since the data traffic is encrypted with the RP Tree session key in the overlay, and with the SN Tree session key within the local LAN, with the two session keys being independent of one another.

The detailed design of Tiered Tree-based Key Management, analysis of its security, and experimental results can be found in [25]. The primary objective in our design is to minimize the amount of key-management control traffic that flows over the satellite links, due to the long delay involved as well as susceptibility to channel errors. We have attempted to ensure that the security of the data traffic does not add any overhead in terms of delay other than that absolutely unavoidable, and that the security protocol does not contribute to deadlocks in group-data dissemination where some group members in certain LANs cannot read the data due to having wrong keys.

From the simulation results, Fig. 10 shows the reduction in key-control traffic over the satellite links using our tiered-tree approach. The graph compares the total key-management

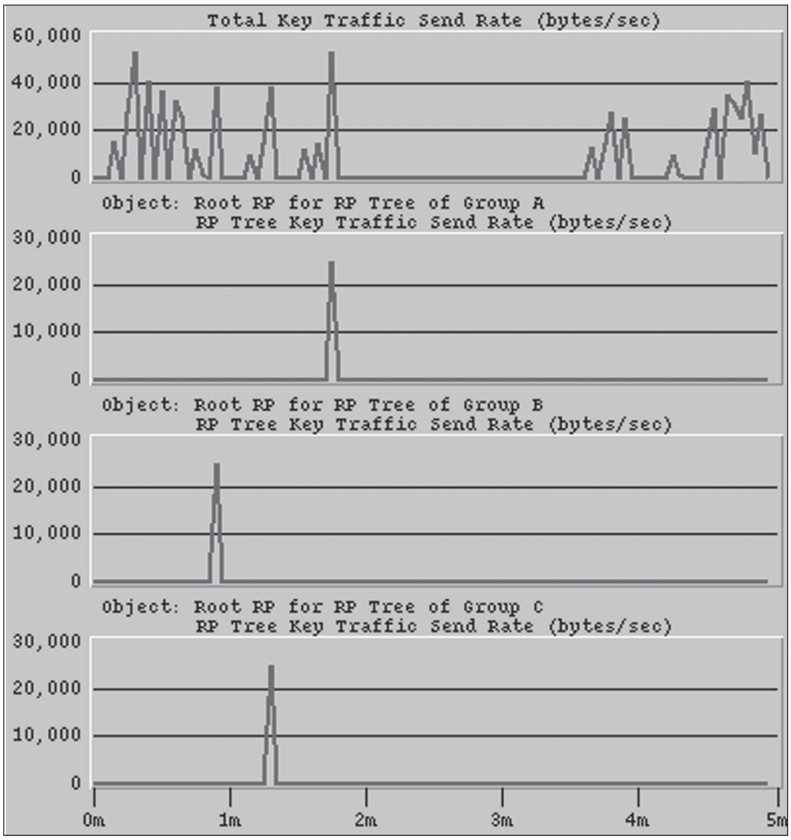


Figure 10. *Tiered tree framework: total key management traffic vs. RP tree traffic for three groups (Y-axis shows the traffic in bytes per second; X-axis is the simulation duration in minutes).*

Our solution is a generic solution aimed specifically at multicast key management and does not deal with an end-to-end security solution for secure communication or give any implementation specifics.

information sent in the network for three simultaneous groups (i.e., sent over the RP trees, sent over the satellite links, and all SN trees limited to local LANs), to the total key information sent on the RP trees (satellite links) only. As the graph shows, the resource savings on the satellite links is substantial when the tiered-tree scheme is used. Even though the group dynamics are high, the amount of message exchanges are very few in the RP tree, that is, over the satellite links. If a flat key-management hierarchy had been used instead, the total key-management traffic would have been sent over the satellite links, thus leading to increased delay and increasing the possibility that the correct keys do not reach all the members at the same time.

Our solution is therefore very scalable. It also acknowledges the fact that the group members might be located in different security domains and, therefore, a single network-wide security management might not be possible. This is a more realistic scenario, since terrestrial LANs might be individual company domains, while the satellite overlay infrastructure is usually owned by a separate entity that provides network connectivity to the LANs, and is not responsible for generating the network traffic. This framework addresses the problem that all users might not be visible to a single, centralized security authority, and the dynamics of user joins or leaves in one LAN should not create an overhead to users in other LANs. Also, in wide-area satellite networks we consider that the satellite channel conditions at a given point in time might be different in different sections of the network. There might be loss in information due to bad channel conditions in some network segments; however, this should not disrupt communication in network segments where the channel conditions are better. Solutions which treat all users in a single tree will not be able to perform as robustly under such conditions. Our solution is also similar to the ML-IPSec concept in that the satellite terminals are only partially trusted; they are allowed to do partial decryption/encryption of the IP packets for efficient routing. However, it is a generic solution aimed specifically at multicast key management and does not deal with an end-to-end security solution for secure communication or give any implementation specifics.

Conclusion

Security is a critical component in hybrid IP-based satellite networks. In this article we have focused on some of the challenges that lie ahead. We have discussed the unique characteristics of hybrid satellite networks that make the problem of ensuring secure communication different from that of purely terrestrial networks. We have presented a survey of the various security solutions that have been proposed, and discussed their advantages and disadvantages. We have proposed several approaches to solve the performance problems of TCP and HTTP in satellite networks arising from secure communication. However, a lot of further work needs to be done to validate our approaches, and we are in the process of developing specific detailed security approaches for typical topologies and validating the proposed designs by simulation. Lastly, we have described our hierarchical approach of key management for providing data security in hybrid networks. We are continuing our research in this area and examining designs to integrate our key-management protocol with the unicast case.

A considerable amount of work needs to be done with regard to secure protocols for hybrid networks, specifically for the case where users are mobile. Here we have touched upon only a small subset of the problems. None of the proposed solutions, including our own, address the question of user authentication or message integrity for group communication. However, we believe the security problems discussed here will receive further treatment from the research community, and this work will be a useful contribution to the field.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The research reported here is supported by the National Aeronautics and Space Administration (NASA) Marshall Space Flight Center under award no. NCC8-235. The views expressed in this article are solely the responsibility of the authors and do not reflect the views or position of NASA or any of its components.

References

[1] J. Border *et al.*, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," IETF RFC 3135, June 2001.
[2] V. Arora *et al.*, "Effective Extensions of Internet in Hybrid Satellite-Terrestrial Networks," University of Maryland, College Park, Tech. Rep. CSHCN TR 96-2, 1996.
[3] V. Bharadwaj, "Improving TCP Performance over High-Bandwidth Geostationary Satellite Links," University of Maryland, College Park, Tech. Rep. ISR TR MS-99-12, 1999.
[4] N. Ehsan, M. Liu, and R. Ragland, "Evaluation of Performance Enhancing Proxies in Internet over Satellite," *Wiley Int'l. J. Commun. Sys.*, vol. 16, Aug. 2003, pp. 513–34.
[5] NIST, "Digital Signature Standard (DSS)," May 19, 1994.
[6] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, Feb. 1997.
[7] R. Atkinson and S. Kent, "Security Architecture for the Internet Protocol," IETF RFC 2401, Nov. 1998.
[8] IETF Transport Layer Security Working Group, "The SSL Protocol Version 3.0," Nov. 1996, available at http://wp.netscape.com/eng/ssl3/draft302.txt
[9] R. Atkinson and S. Kent, "IP Encapsulating Security Payload (ESP)," IETF RFC 2406, Nov. 1998.
[10] Y. Zhang, "A Multilayer IP Security Protocol for TCP Performance Enhancement in Wireless Networks," *IEEE JSAC*, vol. 22, no. 4, 2004, pp. 767–76.
[11] M. Karir and J. Baras, "LES: Layered Encryption Security," *Proc. ICN'04*, Guadeloupe (French Caribbean), Mar. 2004.
[12] E. Olechna, P. Feighery, and S. Hryckiewicz, "Virtual Private Network Issues Using Satellite Based Networks," *MILCOM 2001*, vol. 2, 2001, pp. 785–89.
[13] P. Chitre, M. Karir, and M. Hadjitheodosiou, "TCP in the IPSec Environment," *AIAA ICSSC 2004*, Monterey, CA, May 2004.
[14] SSL Accelerator, Spacenet Inc., available at http://www.spacenet.com/technology/advantages/ssl.html
[15] M. P. Howarth *et al.*, "Dynamics of Key Management in Secure Satellite Multicast," *IEEE JSAC*, vol. 22, no. 2, 2004, pp. 308–19.
[16] C. Wong, M. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," *IEEE/ACM Trans. Net.*, vol. 8, 2000, pp. 16–30.

[17] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast: Issues and Architectures," IETF RFC 2627, June 1999, available at http://www.apps.ietf.org/rfc/rfc2627.html
[18] G. Noubir and L. von Allmen, "Security Issues in Internet Protocols over Satellite Links," *Proc. IEEE VTC '99*, Amsterdam, The Netherlands, 1999.
[19] L. Duquerroy *et al.*, "SatIPSec: An Optimized Solution for Securing Multicast and Unicast Satellite Transmissions," *22nd AIAA Int'l. Commun. Sat. Sys. Conf. and Exhibit*, Monterey, CA, May 2004.
[20] Y. Song, V. Leung, and K. Beznosov, "Supporting End-to-End Security across Proxies with Multiple-Channel SSL," *Proc. 19th IFIP Info. Sec. Conf.*, Toulouse, France, Aug. 2004, pp. 323–37.
[21] A. Roy-Chowdhury and J. Baras, "Key Management for Secure Multicast in Hybrid Satellite Networks," *19th IFIP Info. Sec. Conf.*, Toulouse, France, Aug. 2004.
[22] A. Roy-Chowdhury and J. Baras, "Framework for IP Multicast in Satellite ATM Networks," *AIAA ICSSC 2004*, Monterey, CA, May 2004.
[23] G. Armitage, "Support for Multicast over UNI 3.0/3.1 Based ATM Networks," Internet RFC 2022, Nov. 1996.
[24] S. Deering *et al.*, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Trans. Net.*, vol. 4, no. 2, 1996, pp. 153–62.
[25] A. Roy-Chowdhury, "IP Routing and Key Management for Secure Multicast in Satellite ATM Networks," Master's thesis, University of Maryland, College Park, 2003, available at http://techreports.isr.umd.edu/reports/2004/MS2004-1.pdf

BIOGRAPHIES

AYAN ROY-CHOWDHURY (ayan@isr.umd.edu) received his B.E. in electronics and telecommunications engineering in 1998 from Jadavapur University, India, and his M.S. in electrical engineering in 2003 from the University of Maryland, College Park, where he is currently a Ph.D. student. Between 1998 and 2000 he worked as a senior software engineer at Wipro Technologies, India. His research focuses on the design of protocols and frameworks for secure communication in hybrid networks. He is working on secure protocols for unicast and multicast routing in networks that have wired and wireless terrestrial components interconnected by satellite links. He is also looking into key management techniques for secure data transmission for these network architectures, and efficient user-authentication mechanisms for the same. As part of these topics, he is also investigating performance problems for network communication in satellite networks when security is involved.

JOHN S. BARAS [F] received a B.S. in electrical engineering from National Technical University of Athens, Greece, in 1970, and M.S. and Ph.D. degrees in applied mathematics from Harvard University in 1971 and 1973, respectively. He was founding director of the Institute for Systems Research (one of the first six NSF Engineering Research Centers) from 1985 to 1991. Since August 1973 he has been with the Electrical and Computer Engineering Department and Applied Mathematics Faculty at the University of Maryland, College Park. In 1990 he was appointed to the Lockheed Martin Chair in Systems Engineering. Since 1991 he has been director of the Center for Hybrid and Satellite Communication Networks (a NASA Research Partnership Center). Among his awards are the 1980 Outstanding Paper Award, IEEE Control Systems Society; the 1978, 1983, and 1993 Alan Berman Research Publication Awards, NRL; the 1991 and 1994 Outstanding Invention of the Year Awards, University of Maryland; the Mancur Olson Research Achievement Award, University of Maryland; the 2002 Best Paper Award 23rd Army Science Conference; the 2004 Best Paper Award, 2004 WiSe Conference. He holds three patents. His research interests include wireless networks and MANET, wireless network security and information assurance, integration of logic programming and nonlinear programming for trade-off analysis, multicriteria optimization, noncooperative and cooperative dynamic games, robust control of nonlinear systems and hybrid automata, mathematical and statistical physics algorithms for control and communication systems, distributed asynchronous control and communication systems, object-oriented modeling of complex engineering systems, satellite and hybrid communication networks, network management, fast Internet services over hybrid wireless networks, stochastic systems, planning and optimization, intelligent control and learning, biologically inspired algorithms for signal processing, and sensor networks.

MICHAEL HADJITHEODOSIOU [M] received an M.A. (honours) in electrical and information sciences from the University of Cambridge, United Kingdom, in 1989, an M.S. in electrical and computer engineering from the University of California, Irvine in 1992, and a Ph.D. in engineering (specializing in satellite communications) from the Centre for Satellite Engineering Research (CSER) at the University of Surrey, United Kingdom, in 1995. Among his awards are a scholarship award for studies at the University of Cambridge from the Cambridge Commonwealth Trust (1984–1986); a Fulbright Scholarship for post-graduate work in the United States (1989–1991); a Research Fellowship from the U.K. Engineering and Physical Sciences Research Council (EPSRC) (1992); and the Canadian National Science and Engineering Research Council (NSERC) post-doctoral fellowship award (1995). He worked as a research fellow in the Communication Systems group of CSER (1991–1995) and spent a year as a visiting fellow at the Canadian Government Communications Research Center (CRC) (1995–1996). In November 1996 he joined the Center for Satellite and Hybrid Communication Networks (CSHCN) at the Institute for Systems Research, University of Maryland, College Park, where he is currently an assistant research scientist. He is an expert on space communications and satellite networks. His research interests include performance optimization of wireless and hybrid networks, security and protocol support issues for satellite systems, and design optimization of next-generation broadband satellite networks and applications. He is currently working on supporting the communication needs of NASA enterprises and the communication architecture enabling space exploration. He is currently serving as secretary of the IEEE Satellite and Space Communications Technical Committee.

SPYRO PAPADEMETRIOU received his B.S. in computer science from George Mason University, Fairfax, Virginia. Since then he has been actively involved in Internet research and development within both industry and academia. He was the principal Internet researcher at Synectics Corp., where he developed network and database software. He worked as a researcher at the University of Maryland's Institute for Systems Research, where he designed and developed their first networking laboratory, which is part of the CSHCN. At Inktomi Corp. he spearheaded client acceleration research and was a member the content-distribution network design team. These resulted in several patent filings, of which he holds one. The latter also resulted in American Online's Web client accelerator product. Currently he is with Orbital Data Corp. working on network and application optimization. His research interests include network optimization, application optimization, satellite and terrestrial wireless networking, delay-tolerant networks, sensor networks, distributed systems, and network software architecture.

We have touched upon only a small subset of the problems. None of the proposed solutions, including our own, address the question of user authentication or message integrity for group communication.