On Securely Enabling Intermediary-Based Services and Performance Enhancements for Wireless Mobile Users

Sneha Kasera*
School of Computing
University of Utah
kasera@cs.utah.edu

Semyon Mizikovsky, Ganapathy S. Sundaram and Thomas Y.C. Woo
Bell Laboratories
Lucent Technologies
{smizikovsky, ganeshs, woo}@lucent.com

ABSTRACT

Intermediary-based services and performance optimizations are increasingly being considered, by network service providers, with a view towards offering value-added services and improving the user experience of wireless mobile clients at reduced costs. However, in the presence of an end-to-end security mechanism such as IPsec, it is impossible to offer such services without fully compromising end-to-end security. We propose a new architecture to enable intermediary-based services for wireless mobile users while maintaining an acceptable level of end-to-end security. As a part of our architecture, we present a new IPsec option called Encapsulating Security Variable Payload (ESVP). We identify several important issues related to the architecture and discuss methods for addressing them.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Design, Security

Keywords

Intermediary, Performance, Wireless, Mobile, End-to-End Security, IPsec

1. INTRODUCTION

Mobile users connected to the Internet through wireless links are typically resource limited in terms of end system processing, battery power, and wireless link bandwidth. There is a growing demand for services and performance enhancement mechanisms that address these resource limitations. In order to meet these demands of mobile and wireless users network service providers are increasingly moving from providing just Internet connectivity (a "dumb-pipe") to offering intermediary-based services and performance optimizations to enhance end user experience at reduced costs. Some of these services and performance optimizations include TCP performance enhancements, multimedia packet filtering, header compression, and prevention of Denial-of-Service. These services and optimizations are being, or will be, offered with the help of intermediate nodes placed in the service provider network between communicating end-points. An intermediate node could be a router, a switch, application gateway, a middle box [17], a performance enhancing proxy [5], or a node of an overlay network. In order to provide intermediary-based services and performance optimizations, the intermediate node uses the knowledge of aggregated and per-flow traffic behavior at its location as well as its processing, caching and/or filtering capabilities.

There are two important aspects of the problem of enabling intermediary-based services. First, end users may need to communicate with the network intermediaries for configuration and solicitation of service. Second, the end users must make any information available to the intermediary that might be necessary for them to offer the requested services. The second problem is very challenging especially when an end-to-end security solution such as IPsec is used. The current standard for IP level security (IPsec) enforces the encryption/authentication of the entire payload that is received from the upper layers. Such a function ensures the security of the entire payload, including the transport headers and even network layer headers in some cases, between two end-points that have established a security association [16]. Unfortunately, the current IPsec architecture prevents even trusted intermediaries from examining the payload for providing value added services and performance optimizations. It is possible to use two separate IPsec security associations, one between the end-user and the intermediary and another between the intermediary and the remote end-point. Such a split-IPsec solution is unacceptable to many users because it forces them to expose all their data to the intermediary. Therefore using IPsec, an end-user cannot contract value-added services from a network intermediary unless it fully sacrifices end-to-end security.

In this paper, we address the problem of enabling intermediary-based services for mobile and wireless users while maintaining an acceptable level of end-to-end security. We first present a variety of intermediary-based services and

*This work was performed when the author was with Bell Laboratories.

WiSE'03, September 19, 2003, San Diego, California, USA.

performance enhancements that are beneficial to mobile and wireless users. Next, we propose a new architecture for securely enabling these intermediary-based services. As a part of our architecture, we present a new IPsec option called Encapsulating Security Variable Payload (ESVP) that allows a variable but contiguous portion of the payload to be encrypted/authenticated between the two end-points of a security association and leaves the remaining portion of the payload in the clear. The decision about which portions of the payload should be available as cleartext is taken only by the end-points of the security association. This option allows IPsec to accommodate the tussle between the end-points and the service providers [7], i.e., the service providers want to peek into visible information of the packets for providing value-added services while the end-points decide, based on the benefits they receive, what portion of the information is available to the service providers.

While the ESVP security association determines what portion of the payload is cleartext, this does not necessarily mean that the cleartext is exposed to every intermediary. For example, the cleartext in the IP ESVP packet could be potentially encrypted/decrypted by other protocol layers including another IPsec layer or wireless link layer. The termination points of these layers are trusted intermediaries that are allowed to examine or in some special cases even modify the cleartext for enabling value added services and performance enhancements. It should be noted that although ESVP is described in the context of IPsec, interestingly, it could be implemented at any protocol layer.

Our work on securely enabling intermediary-based services for wireless mobile users is still in progress. The goal in this paper is to identify the architecture and important issues related to our problem.

The rest of the paper is structured as follows: Section 2 contains examples of Intermediary-based services. Our architecture for securely enabling intermediary-based services is described in Section 3. A new IPsec option for exposing information to the intermediary is described in Section 4. In Section 5 we present important issues related to user mobility. In Section 6 we present an example application of our architecture. Section 7 contains a comparison of one-to-many and one-to-one security associations. We summarize the related work in Section 8 and conclude in Section 9.

2. INTERMEDIARY-BASED SERVICES AND PERFORMANCE ENHANCEMENTS

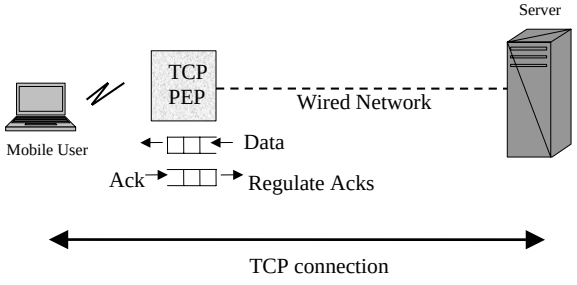

Figure 1: TCP Ack Regulator.

In order to motivate our work we now describe some intermediary-based services and performance optimization mechanisms.

- *TCP Enhancements:* Enhancements to transport protocols such as TCP over error prone and bandwidth-limited links has been an area of study for almost a decade. Particularly, when wireless links are involved, the variance in delay is found to be an important factor influencing TCP performance [6]. Large delay variance decreases the effective client throughput of all TCP-based applications. An accepted mechanism for enhancing TCP performance in such situations is the implementation of a TCP-PEP at an intermediate node. The TCP-PEP can examine, modify or generate TCP packets so as to match the characteristics of the wireline interface to that of the wireless interface thus improving end-to-end TCP performance. Figure 1 shows an example of TCP throughput enhancement for a mobile wireless user. In this figure, the mobile user is communicating with a server using TCP. An intermediate TCP-PEP regulates the acknowledgments [6] from the mobile host to account for the large variations in wireless delay experienced by data flowing towards the mobile, thereby enhancing overall TCP throughput.

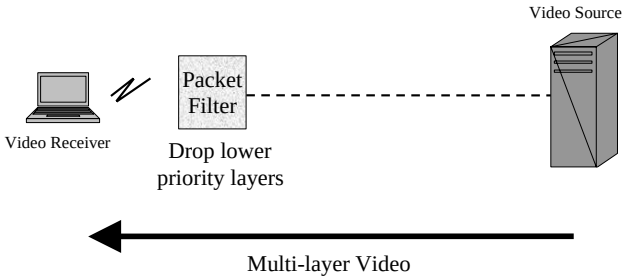

Figure 2: Multimedia Packet Filtering.

- *Packet Classification and Policy Implementation:* An intermediate node could identify flows based on source and destination IP addresses, TCP/UDP source and destination port numbers, and next protocol identity, to offer quality of service guarantees and differentiated treatment to certain packets. For example, the intermediate node, could assign lower priority to non-conforming UDP traffic and a higher priority to TCP traffic during link congestion. A specific classification method and policy implementation depend on the application. Figure 2 shows an example of filtering packets based on multimedia header information. In this figure, multi-layer video is transmitted from the source to a wireless receiver. Based on changing conditions of the wireless link to the receiver, the intermediate node, in the path from the source to receiver, selectively drops packets of lower priority layers. The priority of the layers is found in the multimedia transport header. The intermediate node performing the selective dropping must have the knowledge of the multimedia header format. Keller [15] has demonstrated

dramatic improvements in video quality by using one such scheme.

- *Header Compression:* Compressing protocol headers over wireless links will help save precious wireless bandwidth [13, 10]. Even though, it is possible to achieve header compression between two end-points of an IP tunnel or two adjacent IP hops, most of the header compression schemes are sensitive to delays and loss between the end-points. [8] shows that the average header size increases significantly at high loss. In [9] the authors show the impact of delay on the efficiency of their header compression scheme. Achieving header compression and decompression close to a congested link with the help of an intermediary will help in improving the performance of the header compression schemes. One might argue that if the last hop wireless link is the only congested link that contributes most of the loss and delay then an intermediary-based header compression will not necessarily improve performance over end-to-end header compression. This is not the case when both the end-points are wireless users. We also believe that the single wireless link case will not be true in future multihop wireless networks where multiple bandwidth limited and lossy wireless links might be present. Implementing end-to-end header compression in such situations will result in partial gains only. An intermediary-based header compression, with an intermediary for every wireless link, will immensely help in improving the performance of header compression due to lower loss and delay.

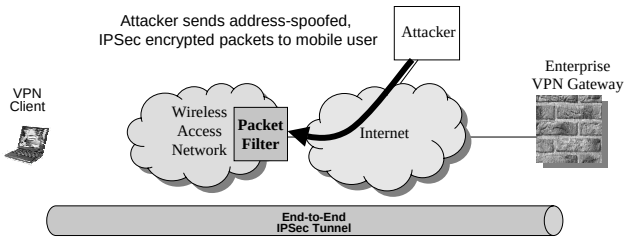

Figure 3: Prevention of Denial-of-Service

- *Prevention of Denial-of-Service:* Intermediate nodes could be configured to filter out packets from unwanted sources to enterprise VPN clients. Enterprise Virtual Private Network (VPN) clients commonly establish secure sessions with their enterprise gateways for accessing their company resources (computers and servers). These clients, especially bandwidth limited wireless mobile enterprise users, can be potentially flooded with unwanted IPsec packets from spoofed enterprise IP addresses. These unwanted packets could be "ingress-filtered" at an intermediate node (e.g., a Packet Data Serving Node (PDSN)) in the path from the enterprise client to the enterprise gateway by setting up an additional authentication tunnel between the enterprise gateway and the intermediate node. On receiving packets with source addresses set to valid enterprise IP addresses, the intermediate node allows only those packets that it can authenticate and drops the rest.

3. ARCHITECTURE

We now present an architecture for securely enabling intermediary-based services. Our architecture has four components.

3.1 Communication between End-points and Intermediary

Communication between end-points and the intermediary may be necessary, in order to *advertise, configure, provision, register, solicit, consent and negotiate* services. For e.g., a network intermediary must be configured to set up additional authentication tunnels for enabling denial-of-service prevention. Even in cases, where an intermediary can transparently perform its services without actively interacting with the end-points, explicit communication between end-points and the intermediary may be required when end-to-end security solutions (e.g., IPsec) are used to set up trust relationships and security associations.

Our architecture includes a protocol that is built on top of a reliable communication channel (using TCP) between the end-points and the intermediary. This protocol is used for all the communication between the end-points and the intermediary mentioned above. In fact this protocol is necessary to securely set up intermediary-based services even when there is no end-to-end security mechanism in place. A broad skeleton of the communication protocol is presented below.

- *Advertisement:* This step of the protocol allows intermediaries to advertise the services and performance enhancements that are offered.
- *Registration:* This step includes the processes of the client choosing the services, as well as mutual authentication between the intermediary and the end-points.
- *Provisioning:* This step of the protocol includes the exchange of all necessary parameters (for the relevant services) as well any session key that the end-points might need to exchange individually with the intermediary.

Our protocol requires the intermediary to be addressable at the IP layer[1].

3.2 Exposing Information

Another extremely important aspect of enabling intermediary-based services is selective exposure of information to an intermediary by the end-points that might be required for offering services. Typically, in order to provide service, an intermediary may need access to the protocol headers of the data packets. For example, an intermediary providing a TCP PEP service [2] will need access to the TCP headers. Currently, there is no standard way of exposing and accessing protocol headers when an end-to-end security protocol such as IPsec Encapsulating Security Payload (ESP) [16] is used.

In our architecture, we propose a new IPsec option called *Encapsulating Security Variable Payload (ESVP)* for selectively exposing information. The details of this new option

[1]This requirement has also been identified in [11].

are described in Section 4. The information that is exposed to an intermediary is secured from the rest of the network by using additional security layers between the end-points and the intermediary. Our architecture allows the flexibility of using additional IPsec layers between the end-points and the intermediary, and any link layer security mechanisms between a wireless user and the intermediary if the intermediary is its link layer peer. In many cases, the wireless link layer security is mandatory and therefore our architecture allows this to be used without incurring additional overheads at the IP layer. The examples described in Section 6 will further clarify this issue.

It should be noted that the service for prevention denial-of-service attacks does not require exposing any information. It only requires communication between the end-points and the intermediary to set up additional authentication tunnels.

3.3 Policy Engine

There are several critical dimensions of the problem of selectively exposing information - *who decides what to expose and whom to expose to*, and *access rights to the exposed information*. What information should be exposed to the intermediary will depend upon the services offered by the intermediary and the security requirements of the end user applications. The question of who has the authority, an end-point or an intermediary, to decide what to expose is extremely important and has serious security implications. Another important question is - *should an intermediary be allowed to only inspect the exposed information but not modify it or should an intermediary be allowed to inspect as well as modify the exposed information?* The answers to these questions will once again be service and/or application specific.

In our architecture we provide a policy engine that generates the rules for addressing the above questions. Although our policy engine is flexible, we believe that the decision of what information should be exposed, to which intermediary, and what access rights to the information are allowed should be decided by the end-points only. Once the rules are made for a particular session (or sessions requiring a certain kind of service) a rule engine at the end-points generates the appropriate ESVP packets for a session.

3.4 Detecting Inappropriate Behavior of the Intermediary

Preserving acceptable security and allowing an intermediary to perform its services, while selectively exposing information to an intermediary is a challenging task. Once again, this aspect of the problem is also multi-dimensional. First, *how much trust could be placed on an intermediary?* The answer depends on the end-user applications and services. Second, *how can one ensure that the intermediary does not play "end-to-end" games*[2] *?* For example, an intermediary with access to TCP headers could change the ordering of TCP packets even when the TCP payload is encrypted. We are considering adding additional fields in the encrypted portion of the ESVP packet to detect any attempts by the intermediary to play "end-to-end" games. The details of the overheads in terms of bytes, how often the additional fields are added (with every packet or statistically with only a small random subset of packets), enhanced rule engines to add these bytes and detect inappropriate behavior at end-points, etc. are still being worked out.

[2]Private communication with Steve Bellovin.

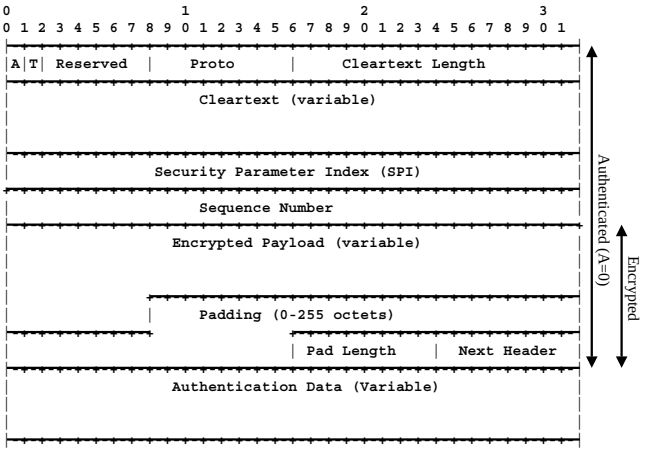

Figure 4: ESVP Packet Format.

4. ENCAPSULATING SECURITY VARIABLE PAYLOAD (ESVP)

We now describe our approach to exposing partial end-to-end packet information to a trusted intermediary. We propose a new IPsec option called ESVP that extends IPsec ESP to obtain more flexibility by leaving out certain portion of the payload in the clear. The cleartext must be a contiguous block from the head or tail of the payload. An ESVP packet has four additional octets in comparison to an ESP packet. Figure 4 shows the format of the ESVP packet. All the fields of the ESVP packet are described below.

- *A:* This is a one bit field, called the A-bit. When the A-bit is set to 0 the cleartext is authenticated end-to-end. Otherwise, the cleartext is not authenticated end-to-end. This field identifies whether an intermediary has permission to only inspect the cleartext of the packet or is it allowed to modify it too.

- *T:* This is a one bit field, called the T-bit. It indicates whether the head or tail of the payload is encrypted. When the tail of the payload is encrypted, the T-bit is set to zero to indicate that the cleartext is placed before the SPI field. When the head of the payload is encrypted, the T-bit is set to 1 and the cleartext follows the Authentication Data field. The T-bit helps in preventing multiple encryptions of the same data as shown in the example in Section 6.

- *Reserved:* These six bits are reserved for indicating any other properties of the cleartext in the future.

- *Proto:* This is a one octet field that indicates the next protocol.

- *Cleartext Length:* This field contains the length in octets of the cleartext. This field is two octets long.

- *Cleartext:* This is the part of the payload received from the upper layers that is left out in the clear.

The rest of the fields of the ESVP packet are same as those of an ESP packet. Detailed definitions of these fields can be found in [16].

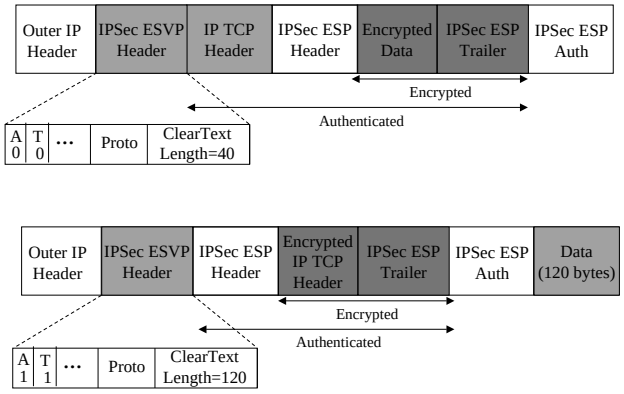

Figure 5: ESVP in Tunnel Mode.

ESVP must be supported in both transport and tunnel mode. Figure 5 shows ESVP in the tunnel mode for a typical IPv4 packet when the inner IP and TCP headers are left in the clear. In both the transport and the tunnel mode the Proto field of the outer IP header should have a new value indicating that the next protocol is ESVP.

4.1 ESVP Security

ESVP relies on ESP for handling the encrypted portion of the payload. Furthermore, it relies on Internet Key Exchange (IKE) [12] for setting up and managing the keys required to perform encryption and authentication. Therefore, the security properties of ESVP with respect to the encrypted portion of the payload should derive directly from the properties of IKE and ESP.

Conversely, the security properties of the portion of the packets that ESVP leaves unencrypted (the cleartext) derive from the properties of the additional mechanism used to secure that portion of the packets. For example, in case ESVP is used to secure the unencrypted portion of the packets between the end points and a trusted intermediary, as with the IP/TCP headers in the case of the example in Section 6, the security properties of ESP will also apply to the IP/TCP headers, albeit allowing a trusted intermediary to have access to them. In this case, no end-to-end security property applies to the unencrypted portion of the packets, since the handling of such portion of the payload is left to security associations which are not end-to-end.

However, it must be noted that with ESVP the end points have complete control over which portion of the packets, if at all, is not encrypted or authenticated. Therefore security policies can be implemented by system administrators who can decide, even on a per-packet basis, to what extent of the packets ESVP should be applied, depending on the trust relationship that they have established with their service providers, as well as on the additional security mechanisms that are available to protect the unencrypted portions of the packets.

4.2 ESVP at Other Layers

We propose ESVP as an IPsec option to offer a generic capability at the IP layer that can be used by all the layers above IP. ESVP could be implemented at other layers too by appending the first four bytes of ESVP packet format to the packet at any layer. For e.g., if ESVP were to be implemented at the secure socket layer (SSL), the first four bytes of ESVP packet format are appended to the partially encrypted application data, where the cleartext length of ESVP now refers to any contiguous unencrypted application data. By implementing ESVP at the socket layer, it is possible to enable intermediary-based overlay services.

5. IMPACT OF MOBILITY

In this section we present two important issues that highlight the impact of user mobility on intermediary-based services. As we have done so far in the paper, for the simplicity of presentation, we assume that a single intermediary provides services to a session between a wireless mobile user and an enterprise gateway. The following discussion applies to more general scenarios, including those involving multiple intermediaries, as well.

5.1 Intermediary Communication

The basic communication protocol for enabling intermediary-based services was outlined in Section 3.1. However when a client moves from one network to another, the handoff procedures may require a change in the intermediary. This applies to both *idle roaming* as well as *active handoff*. In idle roaming a client travels to a foreign network. In active handoff a client moves into a foreign network during an active session. In either case, the *advertisement* and *registration* procedures must be repeated. Recall, that *registration* involves a mutual authentication procedure by which the intermediary and the end-points confirm each others' identity. Following this, the intermediary receives an authorization to provide one or more services. This authentication and authorization step involves communicating with the home network.

5.2 Key Management

Management of the keys shared with the intermediary becomes extremely important during active handoff. During active handoff, the security association between the two endpoints should not be affected. However, for the security associations between the end-points and the intermediary, two choices exist: a secure session key transfer between old and the new intermediary, or formation of new security associations (requiring new keys) with the new intermediary. In the event the home network is involved in the registration process, it may be efficient to securely transfer the keys that are used in the old security association between the mobile user and the intermediary, from the old intermediary to the new one. In particular, this will be applicable to the case of link layer handoff where existing wireless standards already address this issue [1]. However, a transfer of keys, transfer of buffered data as well as any cryptographic counter information (in order to prevent replay attacks) between the intermediaries will require a secure path between them.

For securing the path between the intermediary and the other end-point, we propose creating a new security association between the new intermediary and the other end-point. When the other end-point is an enterprise gateway it makes

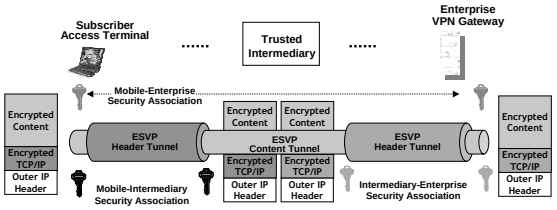

Figure 6: Mobile Wireless Enterprise User Benefiting from Intermediary-based Services.

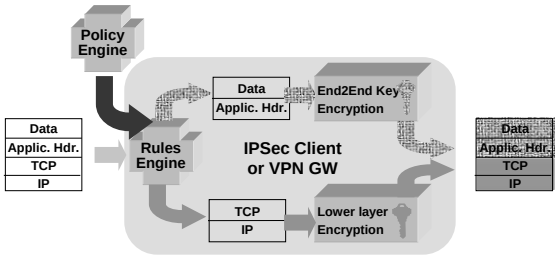

Figure 7: ESVP Packet Generation.

sense to multiplex multiple sessions in one security association between the intermediary and the gateway. When this is done, transfer of the keys of the old security association between the intermediary and the enterprise gateway to the new intermediary may result in compromising the security of all the other sessions that are multiplexed on the old security association. Our proposal for establishment of a new security association between the new intermediary and the enterprise server is motivated by this fact.

6. EXAMPLE APPLICATION

In this section we provide an example application that demonstrates some of the important features of our architecture.

Figure 6 shows a mobile wireless enterprise user communicating with an enterprise gateway. As a first step the mobile user goes through the registration, authentication and authorization processes with the wireless service provider (and/or Internet service provider) and the enterprise gateway. If the mobile user is in a foreign domain the home domain is involved in these processes. The mobile user learns about the intermediary-based services (say TCP PEP service) from *advertisement* messages from the intermediary (e.g., a Packet Data Serving Node (PDSN)). The mobile user, the enterprise gateway and the intermediary agree on the services required for the session.

Next, the mobile user establishes an ESVP security association with the enterprise gateway. It uses the secret key exchanged with the enterprise gateway to encrypt TCP payload but leaves the IP/TCP header in the open. In order to secure the IP/TCP header from the rest of the network, the mobile user establishes another ESVP security association with the intermediary and the intermediary establishes a third ESVP security association with the enterprise gateway. In the first ESVP operation at the mobile user, the IP/TCP headers are left in the open and the T-bit is set to 0 because the tail of the IP/TCP packet is encrypted. In the second ESVP operation, the inner IP/TCP headers are encrypted and the T-bit is now set to 1. There is no need to re-encrypt the TCP payload. IPsec ESP security associations could also be used between the end-points and the intermediary but the use of T-bit saves an additional encryption. On receiving the encrypted packet from its security association with the mobile user, the intermediary decrypts the outer ESVP packet and hence the IP/TCP header, performs the TCP PEP service, applies another ESVP operation to secure the IP/TCP header, and sends the encrypted packet using the security association with the enterprise gateway. The enterprise gateway receives the encrypted packet and applies two ESVP operations to obtain the payload and the protocol headers. Note that the outer ESVP tunnel between the mobile and the intermediary could be replaced by a secure wireless link layer when available. This would save the overheads of the second ESVP operation.

Figure 7 shows the ESVP operations together with the policy and rules engine. If the mobile user roams into a foreign domain during an ongoing session, then the handoff procedure will involve either a session key transfer or a new session key setup. However, the end-to-end ESVP security association that exists between the mobile user and the enterprise gateway will remain intact. The mobile user must receive new service advertisements from the intermediary in the foreign domain and agree on the services required. Upon session completion, all the three security associations between the mobile user, the intermediary and the enterprise gateway are terminated.

7. ONE-TO-MANY VERSUS MULTIPLE ONE-TO-ONE SECURITY ASSOCIATIONS

In our architecture, all security associations involving the end-points and the intermediary are one-to-one, i.e., only two nodes (end-point or intermediary) are part of any security association. Alternately, as proposed in [18], it is possible to have, composite security associations or one-to-many security associations that involve more than two nodes, e.g., both the end-points and the intermediary. We now compare the advantages and disadvantages of these approaches. In the comparison below, we consider the general case where multiple intermediaries might be involved in providing services to two end-points.

- A one-to-one security association between two intermediaries (if any), or between an intermediary and a gateway or server end-point, will allow multiplexing several sessions into one security association whereas the one-to-many approach will require as many security associations as the number of users.
- In the case of the one-to-one approach, user mobility involving only its first intermediary will not affect the security associations among the other intermediaries and the security associations between the intermediaries and the other end-point. With the one-to-many approach, every change of an intermediary will affect all the nodes.
- One potential advantage of one-to-many security associations is that since the same key is used across all the nodes involved in the security association an intermediary must not necessarily decrypt and encrypt every packet. In the case of one-to-one security associations,

each security association has a separate key. Therefore, an intermediary must decrypt using one key and encrypt using another key every packet that is transmitted end-to-end.

- A one-to-many security association allows more generality in terms of making different parts of the packet accessible to a different subset of intermediaries. The one-to-one approach will require a large number of security associations to achieve this.

We chose the one-to-one security associations because they are simpler, use well-established one-to-one key exchange mechanisms, are more efficient in the presence of user mobility, and will address most of the intermediary-based services that we envision.

8. RELATED WORK

Several intermediary-based services have been proposed and studied extensively (e.g., [2, 5, 6] for TCP performance enhancements over wireless, [14, 15] in the context of active networks, OPES [3], MIDCOM [17]). None of the above proposals address the issue of how could packet level information be made available to an intermediary when a security solution such as IPsec is used.

In [4], Bellovin proposed a variant of ESP called Transport-Friendly ESP (TF-ESP) which allowed for leaving out certain portions of the payload in the clear. He also suggested that the cleartext be integrity protected with the rest of the ESP header. The problem with this approach is that when the ESP header is integrity protected with keys known only to end-points, the intermediaries cannot verify if the information is correct. Also, the end-to-end integrity protection does not allow an intermediary to enable those services or performance enhancements that require modification of the cleartext. In ESVP, the A-bit allows an endpoint to selectively grant the trusted intermediary, read-only or read/write access to the cleartext. We propose to address the problem of verification of cleartext (whether authenticated end-to-end or not) at the trusted intermediary by using another ESVP tunnel between the end-point and the trusted intermediary. ESVP also allows the flexibility of having the head or tail of the payload in the clear which prevents double encryption for certain applications. The example in Section 6 used this flexibility. Another situation where the T-bit could prevent additional encryption of encrypted data is when Secure Socket Layer (SSL) is used over ESVP.

Bellovin proposed another variant of ESP called "Disclosure" Header where all fields of interest are copied from the payload into an unencrypted portion of the ESP header [4]. Although cleaner, this approach requires pre-defined header formats to be known to the trusted intermediaries and endpoints, making it less flexible. The trusted intermediaries also need to be informed about which "disclosure" header format is being used. This approach also increases the length of the packet which might be prohibitive for bandwidth limited wireless scenarios.

In [18], Zhang *et al* have proposed a very generic approach called Multi-Layer IPsec (ML-IPsec) that divides the payload into multiple zones such that each zone could be encrypted with a different key. Composite security associations involving intermediaries are established and intermediaries with the keys to encrypt/decrypt certain zones are allowed access to those zones. The fine-granular control provided by this approach makes it somewhat complex. Especially, ML-IPsec changes the nature of the security associations from one-to-one to one-to-many. We retain the security association as one-to-one.

SSL secures only the application payload and leaves out the transport and network layer headers as cleartext. Therefore SSL over IPsec could be used to obtain some intermediary-based services such as TCP PEP. Our approach proposes a simple framework that does not restrict the services to be based only on exposure of IP/TCP headers. Our framework could be applied at the IP layer or above for a variety of services including those that expose application headers.

Last, but very important from our viewpoint, neither of [4, 18] deals with mobility related issues or with issues related to dynamic invocation and revocation of intermediary-based services. Our architecture specifically addresses wireless mobile users. It also allows dynamic invocation and revocation of intermediary-based services.

9. CONCLUSIONS

We have proposed a new architecture for securely enabling intermediary-based services for wireless mobile users. Currently we are working on the detailed design, implementation, and evaluation of this architecture.

10. ACKNOWLEDGEMENTS

We thank I. Faynberg, E. Grosse, A. Mankin, S. Miller, R. Ramjee, and L. Salgarelli for many valuable discussion and support for this work.

11. REFERENCES

[1] Guide to 3rd Generation Security. TR 33.900, Third Generation Partnership Program 2 (3GPP2).

[2] H. Balakrishnan, S. Seshan, and R. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, December 1995.

[3] A. Barbir, R. Chen, M. Hofmann, H. Orman, and R. Penno. An Architecture for Open Pluggable Edge Services (OPES). Internet Draft, December 2002. draft-ietf-opes-architecture-04.txt.

[4] S. Bellovin. Transport-friendly ESP (or Layer Violation for Fun and Profit). IETF-44 TF-ESP BOF, March 1999.

[5] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135, June 2001.

[6] M. Chan and R. Ramjee. TCP/IP Performance over 3G Wireless Links with Rate and Delay Variations. In *Proc. of ACM Mobicom*, September 2002.

[7] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. In *Proc. ACM SICOMM Conference*, August 2002.

[8] M. Degermark, H. Hannu, L. Jonsson, and K. Svanbro. Evaluation of cRTP Performance over Cellular Radio Links. *IEEE Personal Communications*, pages 20–25, August 2000.

[9] S. Dorward and S. Quinlan. Robust Data Compression of Network Packets, 2000. http://www.cs.bell-labs.com/cm/cs/who/seanq/networkcomp.pdf.

[10] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. Robust Header Compression (ROHC): Framework and Four profiles: RTP, UDP, ESP, and uncompressed. RFC 3095, July 2001.

[11] S. Floyd and L. Daigle. IAB Architectural and Policy Considerations for Open Pluggable Edge Services. RFC 3238, January 2002.

[12] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, November 1998.

[13] V. Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. RFC 1144, February 1990.

[14] S. Kasera, S. Bhattacharyya, M. Keaton, D. Kiwior, J. Kurose, D. Towsley, and S. Zabele. Scalable Fair Reliable Multicast using Active Services. *IEEE Networks*, January 2000.

[15] R. Keller, S. Choi, D. Decasper, M. Dasen, G. Fankhauser, and B. Plattner. An Active Router Architecture for Multicast Video Distribution. In *Proc. of IEEE Infocom*, March 2000.

[16] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, November 1998.

[17] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan. Middlebox Communication Architecture and Framework. RFC 3303, August 2002.

[18] Y. Zhang and B. Singh. A Multi-Layer IPsec Protocol. In *Proc. 9th Usenix Security Symposium*, August 2000.