Anonymous Internet Communication based on IPSec
Song, Ronggong; Korba, Larry

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC-CNRC

*Anonymous Internet Communication based on IPSec.**

Song, R.

August 2002

* published in: Proceedings of the International Federation for Information Processing (IFIP) World Computer Congress Montreal, Quebec. August 25-30, 2002. NRC 44943.

Canada

Anonymous Internet Communication based on IPSec

Ronggong Song   Larry Korba
Institute for Information Technology,
National Research Council of Canada
E-mail: {Ronggong.Song, Larry.Korba}@nrc.ca

Abstract: Network approaches for anonymous communication have been extant for some time. Unfortunately, there are limitations with these approaches. In this paper, we first expose the limitations of existing anonymous communication networks. We then present an anonymous Internet communication technique based on IPSec. Our technique provides bi-directional, real-time anonymous Internet communication that is resistant traffic analysis for any TCP/IP applications. We describe the signaling protocols and the implementation of our architecture. Our technique is easily implemented in security gateways or IP router that supports IPSec.

Keywords: Internet security, privacy, anonymous Internet, traffic analysis.

1. INTRODUCTION

Privacy is becoming a critical issue on the Internet. Users feel that one of the most important barriers to using the Internet is the fear of having their privacy violated. Governments around the world have introduced legislation placing requirements upon the way in which personal information is handled. In attempt to provide some technical solutions within the privacy void, several network-based privacy-enhancing technologies have been developed in recent years. Some examples of these technologies include: MIX-Network [1, 2], Onion Routing [3, 4], Crowds System [5], Freedom Network [6], etc.

It is a difficult to achieve anonymity for real-time services in the Internet (e.g. Web Access). Some limitations of the above networks are described in [7]. In addition, another important limitation is they are not implemented at the IP layer in such a way that some important security protocols cannot be available, for example, they all cannot support IPSec [8], and furthermore, the Crowds System cannot support SSL [9] at the endpoints. However, IPSec and SSL have become important security protocols, especially for e-commerce applications. Thus, an anonymous Internet at the IP layer, which can support any TCP/IP applications, becomes desiderata.

Our anonymous Internet proposal is based on the IPSec tunneling technique. There are two main reasons. First, the IPSec technique has become a popular security technique, especially for VPN application, and many security gateways, firewalls and routers support IPSec technique. Another reason is that it enables our proposal to support all TCP and UDP applications.

Our principles are as follows. First, our anonymous Internet comprises a set of anonymous Internet nodes (AINs) that support IPSec, such as security gateways, firewalls and routers, etc. The connection between two neighboring AINs is supported by the IPSec tunneling technique and built previously. We call the connection a permanent virtual tunnel (PVT). A creating signaling protocol then copies the PVT as a temporary virtual tunnel (TVT) with a different security parameter index (SPI), and creates an anonymous IP-datagram virtual tunnel (AIVT) from the entrance node to the exit node by connecting these TVTs. The signaling is transmitted through the PVTs. A nested symmetrical encryption channel is established by employing public key system during the AIVT. The end-to-end data then is encrypted using the nested symmetrical encryption algorithm and sent through the AIVT. Final, a destroying signaling protocol is used to destroy the AIVT and TVTs after the session.

The other advantages for our technique are as follows. First, unlike other anonymous communication networks, it is independent of all applications over IP. Unlike Onion Routing and MIX-Network, packet loss causes a problem of network backlog and cascading retransmits, since the nodes talk to each other via TCP in these networks. The end-to-end TCP is not available in them. On the other hand, unlike Freedom Network, in our anonymous Internet architecture, the source address and destination address of the user's data are changed through every node, but every node only needs one public IP address because we use the different IP tunnels to distinguish the different end-to-end IP-datagram streams. Thus, it supports both IPv4 and IPv6.

The rest of the paper is organized as follows. Background describing IPSec and anonymous communication networks are briefly reviewed in the next section. In Section 3, some notations are defined. In Section 4, anonymous IP routing protocols are proposed based on IPSec, including the creating PVT and AIVT protocols, and destroying AIVT protocol. In Section 5, the implementation of the anonymous Internet is concisely described. In Section 6, the vulnerabilities in our architecture are discussed. In Section 7, some concluding remarks and directions are presented for further research.

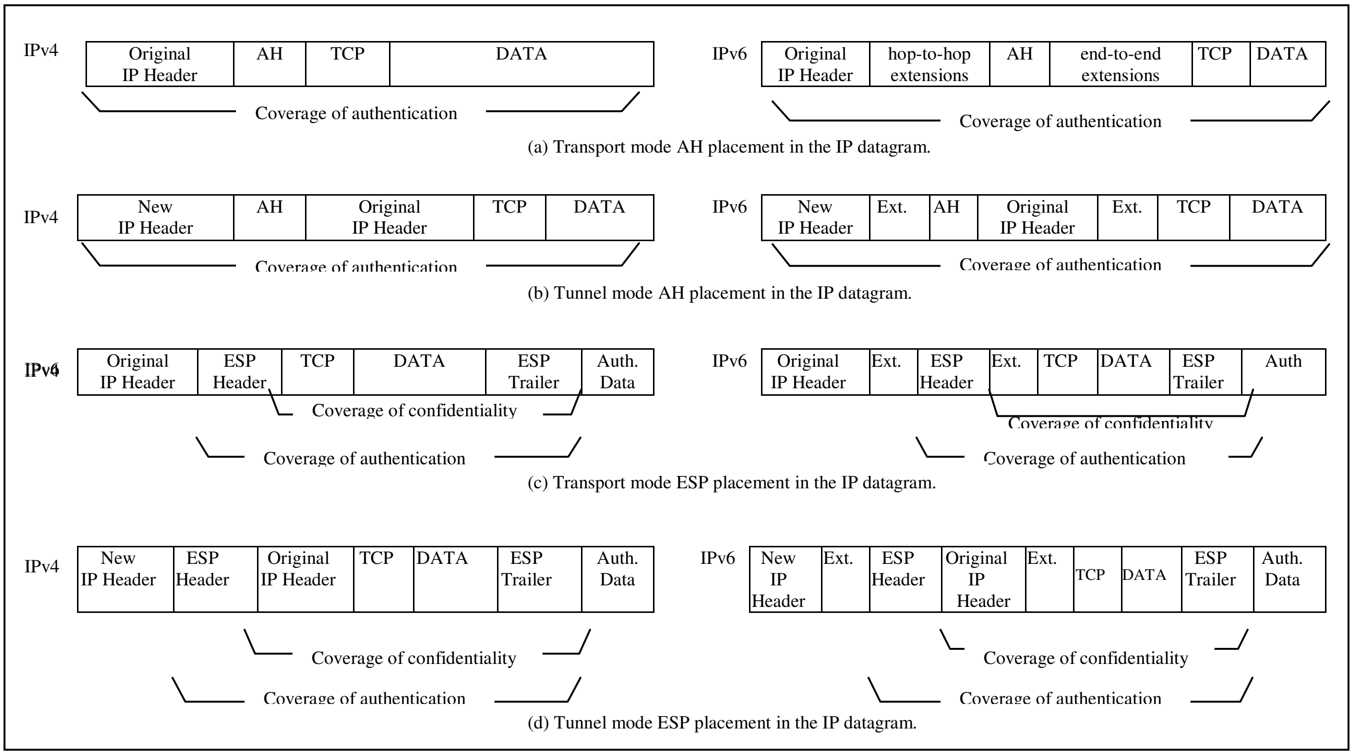
2. BACKGOUND

2.1 IPSec Architecture

The IPSec architecture is the most advanced effort in the standardization of Internet security. IPSec can be used to protect an IP layer path between a pair of end-systems or hosts, between a pair of intermediate systems - called security gateways.

IPSec consists of the following components:

- Two security protocols: the IP Authentication Header (IP AH) [10] and the IP Encapsulating Security Payload (IP ESP) [11] that provide the basic security mechanisms within IP;
- Security Associations (SA) that present the set of security services and parameters negotiated on each security IP path;
- Algorithms for authentication, encryption and integrity.



(a) Transport mode AH placement in the IP datagram.

(b) Tunnel mode AH placement in the IP datagram.

(c) Transport mode ESP placement in the IP datagram.

(d) Tunnel mode ESP placement in the IP datagram.

Figure 1. IPSec Transport and Tunnel Modes.

IP AH and IP ESP may be applied alone or in combination with each other. Each protocol can operate in one of two modes: transport mode or tunnel mode. In transport mode, the security mechanisms are applied only to the upper layer data, and the information contained in the IP header is left unprotected. In tunnel mode, both the upper layer data and the IP header are protected. Figure 1 depicts these modes.

IP AH provides data origin authentication, data integrity and replay detection for IP datagrams. Except for these functions, IP ESP also provides data confidentiality services. But unlike the AH authentication data field, the authentication covers only the ESP header, the ESP payload and the padding fields of the IP datagram in the ESP data field. The IP header is never protected by the ESP authentication service. Thus, in the cases where the data integrity and data confidentiality of the entire IP datagram are required, it would be better to use IP ESP+AH.

A Security Association represents an agreement between two IP nodes on a set of security services to be applied to the IP traffic stream between these nodes. Each SA is associated with AH or ESP services but not both. IPSec provides end-to-end security and VPN services.

2.2 Anonymous communication networks

The primary goal of the anonymous communication network is to protect user anonymous communication against traffic analysis. Chaum first proposes an anonymous communication network: MIX-Network, for supporting anonymous e-mail services.

Based upon Chaum's MIX-Network, Dai [12] has described a theoretical architecture that would provide private protection against traffic analysis based on a distributed system of anonymous packet forwarders. He calls it Pipenet. Pipenet consists of a cloud of packet forwarding nodes distributed around the Internet, and packets from a client would be multiply encrypted and flow through a chain of these nodes. Pipenet is an idealized architecture and has never been built. Pipenet's mortal disadvantage is that its packet loss or delay is extremely large.

Like Pipenet architecture, Onion Routing provides a more mature implementation against traffic analysis. It also provides bi-directional real-time communications. Its disadvantages include that packet loss causes a problem, and does not support the security services such as IPSec, VPN, etc. Freedom Network is another similar technique. It works in a very similar way as compared to the Onion Routing.

The Crowds System is based on a very different principle. A user sends a message with a certain probability into the Internet. Otherwise he forwards the messages to another randomly selected user. That user does the same things and so on. Unfortunately, the Crowds System doesn't support the security services such as IPSec, SSL, etc.

The above networks are not implemented at the IP layer in such a way that their applications have some limitations. Proposals for anonymous Internet at IP layer are in process now. Our proposal is to use the maturing IPSec techniques for supporting anonymous communication at IP layer.

3. TERMINOLOGY

Notations used in the paper are defined as follows.

- *AIN:* Anonymous Internet Node. It is a node such as security gateway, firewall or router supporting IPSec tunneling technique, converting the IP header of the IP datagram into another IP header for hiding the original IP header, and forwarding the data stream from one IP tunnel to another IP tunnel.
- *PVT:* Permanent Virtual Tunnel. This is an IP ESP+AH tunnel between two AINs that is created previously by Internet Key Exchange protocol (IKE) or manual. It is a long-term tunnel depending on its security policy, and supports signaling services for our anonymous Internet.
- *TVT:* Temporary Virtual Tunnel. It is an IP AH tunnel between two AINs, and has the same parameters as the IP AH of the PVT except for SPI. It is a part of an AIVT, and supports data forwarding services for the user's datagrams.
- *AIVT:* Anonymous IP-datagram Virtual Tunnel. It comprises several TVTs that form the AIVT from the entrance AIN to the exit AIN. Note that an AIVT may include several UDP and TCP sessions with the same end-to-end IP addresses. A new AIVT is created for a new end-to-end IP-datagram stream by our creating AIVT protocol. An old AIVT is destroyed after a destroying signal is sent or the AIVT is expired.
- *SPI:* Security Parameter Index. It is a random value used in combination with the destination IP address to identify the security association for that datagram.
- $E_{PK_i}(K_i)$: The symmetrical key is encrypted with the AIN$_i$'s public key PK_i, e.g. RSA.

- $E_{K_i}(M)$: The message *M* is encrypted with the symmetrical key *K*$_i$, *e.g. DES.*

4. ANONYMOUS IP ROUTING PROTOCOLS

4.1 Anonymous Internet Topology

An anonymous Internet consists of some AINs. The connection between two AINs is a PVT created by the IKE protocol or manual previously. Every AIN only accepts the data stream from its customers and some AINs that have a PVT or TVT with this AIN. The AIN then forwards the data stream to the next AIN according to the routing information. An anonymous Internet routing protocol would be a desired approach for this system (e.g. peer-to-peer technology). This needs further research. Topology for anonymous Internet is illustrated in Figure 2.

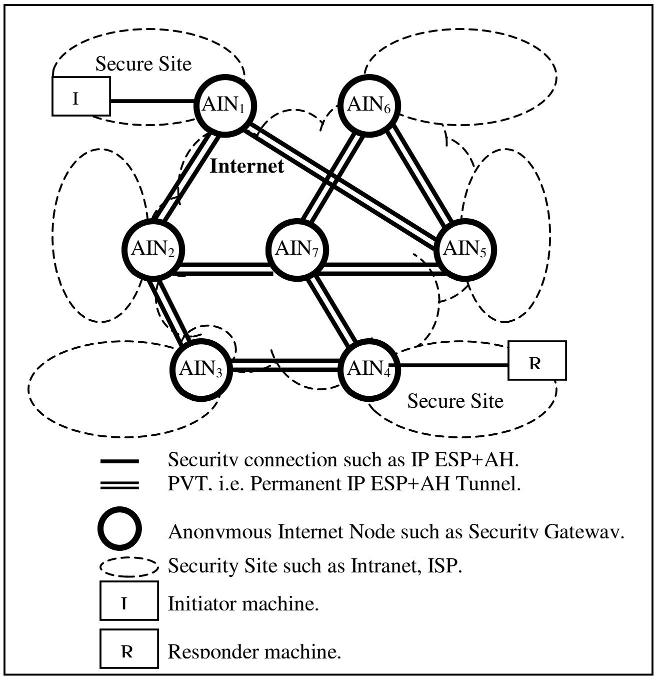

Secure Site
I
AIN_1
AIN_6
Internet
AIN_2
AIN_7
AIN_5
AIN_3
AIN_4
R
Secure Site
Securitv connection such as IP ESP+AH.
PVT. i.e. Permanent IP ESP+AH Tunnel.
Anonvmous Internet Node such as Securitv Gatewav.
Security Site such as Intranet, ISP.
I
Initiator machine.
R
Resvonder machine.

Figure 2. Anonymous Internet Topology.

The anonymous Internet allows the connection between the initiator and the responder to remain anonymous. We call it an anonymous IP tunnel connection. The anonymous connections hide that who is connected to whom, and for what purpose, from both the outside eavesdroppers and compromised AINs. In addition, if the anonymity is also desired at the application layer, all identifying information must be removed from the data stream at the upper layers before being sent over the anonymous IP tunnel connection.

To begin an anonymous session, the initiator sends his/her IP datagrams to the registered AIN using a security connection such as IP ESP+AH tunnel. We call the registered AIN an entrance AIN. The entrance AIN then encrypts the original IP datagrams using the nested encryption described in the Section 5, where the symmetric keys are distributed during the AIVT. According to the destination IP address of the original IP header, the AIN encapsulates the IP datagrams using the IP AH tunnel and sends it to the next AIN if an old AIVT has existed for the source and destination addresses. Otherwise, the AIN creates a new AIVT using a creating AIVT protocol, and then sends the IP datagrams to the next AIN along the AIVT. Final, the AIVT is destroyed by a destroying protocol if the initiator or responder sends a destroying signal after the session, or it is automatically destroyed according to the predefined expiration time.

4.2 Creating PVT Protocol

Based on IETF RFC 2401, the concept of a security association (SA) is fundamental to IPSec. A SA is unidirectional in that it defines the services applied to the IP datagrams transmitted in one direction between the pair of nodes. Both AH and ESP make use of security associations.

To provide secure, bi-directional communication between two AINs, a PVT must comprise four security associations. To make the implementation simple, two of them can use the same parameters except for the destination IP address for IP ESP, and another two for IP AH. An IKE protocol can be used to create the two IP AH tunnels and IP ESP tunnels — a PVT between two AINs.

Note that a PVT is a long-term connection between two AINs, and only provides the signaling transmission services. It doesn't provide the user's data transmission services.

4.3 Creating AIVT Protocol

Every AIVT starts at an entrance AIN, ends with an exit AIN, and passes through several intermediate AINs. The entrance AIN provides the creating function for the AIVT because it is unrealistic to have the user create a suitable route by himself/herself. Actually, sometimes the entrance AIN may only know a part of the whole anonymous Internet topology.

In our anonymous Internet, only the entrance AIN knows the AIVT, and other AIN only knows its previous and next AINs that form the AIVT. Thus, the entrance AIN must be a trusted AIN for the initiator. Our creating AIVT protocol is described as follows.

① The entrance AIN first identifies a series of AINs forming a route through the anonymous Internet, and constructs a creating AIVT signal according to the source and destination IP addresses of the initiator's IP datagram. An anonymous Internet routing protocol would be required. This will be covered in a future paper. Assuming the route consists of AIN_1, AIN_2, …, AIN_n where AIN_1 is the entrance AIN and AIN_n is the exit AIN, Figure 3 depicts the creating AIVT signal.

$$\begin{array}{l}
E_{PK_2}(K_2)\ E_{K_2}(\ \text{AIN}_3\text{, Exp-Time,} \\
\qquad E_{PK_3}(K_3)\ E_{K_3}(\ \text{AIN}_4\text{, Exp -Time,} \\
\qquad\qquad \ldots\ldots \\
\qquad\qquad E_{PK_{n-1}}(K_{n-1})\ E_{K_{n-1}}(\ \text{AIN}_n\text{, Exp-Time,} \\
\qquad\qquad\quad E_{PK_n}(K_n)\ E_{K_n}(\ \textit{NULL, Destination IP Address,}\ \text{Exp-Time,}))\ldots)
\end{array}$$

Figure 3. Creating Signal.

② The entrance AIN then makes a new random $\text{SPI}_{1,2}$, and sends the new $\text{SPI}_{1,2}$ and the above creating signal to AIN_2 through the $\text{PVT}_{1,2}$ between AIN_1 and AIN_2. AIN_1 and AIN_2 then copies the IP AH of the PVT as an new $\text{TVT}_{1,2}$ but using the new $\text{SPI}_{1,2}$. AIN_1 stores K_n, K_{n-1}, …, K_2 as the nested encryption keys for the forward data stream, and the nested decryption keys for the backward data stream.

③ AIN_2 decrypts the creating signal using its private key SK_2, and stores the symmetric key (K_2). AIN_2 then makes a new random $\text{SPI}_{2,3}$ and creates a new $\text{TVT}_{2,3}$ between AIN_2 and AIN_3 as the step ②. Final, AIN_2 creates a bi-directional connection between $\text{TVT}_{1,2}$ and $\text{TVT}_{2,3}$, and uses K_2 as the decryption key for the forward data stream and the encryption key for the backward data stream over the connection.

④ All intermediate AINs act as the step ③.

⑤ Final, AIN_n decrypts the creating signal using its private key SK_n, and stores K_n as the decryption key for the forward data stream and the encryption key for the backward data stream.

Thus, the AIVT between the entrance AIN and the exit AIN is established using the protocol described above for the initiator and responder. Since the data confidentiality is supported by the nested encryption operation at the entrance AIN, IP AH is enough to provide the secure protection for the TVT. Figure 4 depicts an AIVT.

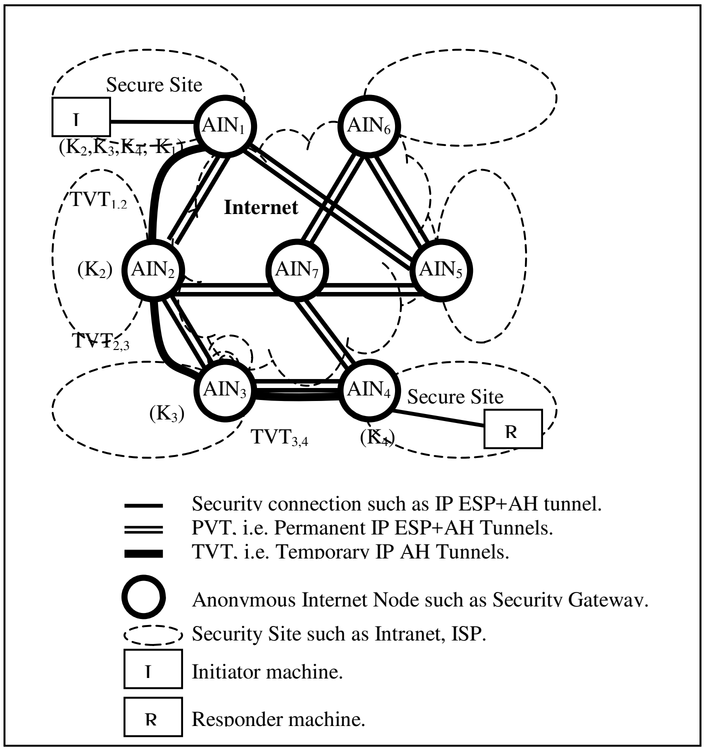

Secure Site
I
AIN1
AIN6
(K2,K3,K4, K1)
TVT1,2
Internet
(K2)
AIN2
AIN7
AIN5
TVT2,3
AIN3
AIN4
Secure Site
(K3)
TVT3,4
(K4)
R
Securitv connection such as IP ESP+AH tunnel.
PVT. i.e. Permanent IP ESP+AH Tunnels.
TVT. i.e. Temporarv IP AH Tunnels.
Anonvmous Internet Node such as Securitv Gatewav.
Security Site such as Intranet, ISP.
I
Initiator machine.
R
Responder machine.

Figure 4. Anonymous IP-datagram Virtual Tunnel.

4.4 Anonymous End-to-end Bi-direction Real-time Data Transmission Protocol

In our architecture, we assume that there is an IP ESP+AH tunnel between the initiator and the entrance AIN, and also between the exit AIN and the responder. The advantages are as follows. First, the entrance AIN and exit AIN can support the data origin authentication for the initiator and responder, respectively. On the other hand, it can hide the responder's IP address from the outside eavesdroppers between the initiator and the entrance AIN, and the initiator's IP address from the outside eavesdroppers between the exit AIN and the responder.

In addition, based on whether the initiator hides its IP address from the exit AIN and the responder or not, two options are available for providing different degree anonymous bi-directional data transmission. The initiator can choose the best one according to its requirements for privacy and security.

The first option is that the initiator hides its IP address from any entities except for the entrance AIN. In this situation, the initiator can get higher degree anonymity protection. The second option is that the initiator doesn't hide its IP address from the exit AIN and the responder. In this situation, the advantage is that it also can support some end-to-end applications at IP layer, for example an anonymous end-to-end IPSec and VPN, etc. The difference is that the initiator chooses a random private IP address as the source address of the original IP-datagram in the first situation, and a real IP address as the source address in the second situation. The technique is described as follows.

① The initiator first prepares his/her original IP-datagram according to his/her option for the anonymous degrees, i.e. using a random private IP address or real IP address as the source IP address of the original IP datagrams, and then sends the original IP datagrams to the entrance AIN using an IP ESP+AH tunnel ($PVT_{i,1}$) between the initiator and the entrance AIN.

② The entrance AIN verifies whether the initiator is its registration customer using IP AH and gets the real IP source address, and then decrypts the data payload using IP ESP and gets the original IP header. The entrance AIN then creates an $AIVT_{1,2}$ using the above creating protocol according to the real source address and original IP header, makes a connection between the $PVT_{i,1}$ and the $AIVT_{1,2}$, and encrypts the whole IP datagram using the nested encryption technique, i.e. the IP datagram first is encrypted with K_n*, then* K_{n-1}*, …, and final* K_2. Final, the entrance AIN creates a new IP AH header for the encrypted data payload according to the $TVT_{1,2}$ and puts the new IP datagram to the $TVT_{1,2}$.

③ AIN_2 verifies the IP AH and decrypts one layer of the encryption data payload using its decryption key K_2, and then creates a new IP AH header according to the $TVT_{2,3}$, and sends the new IP datagram to the next AIN.

④ All intermediate AINs act as the step ③.

⑤ Final, AIN_n verifies the IP AH, decrypts last layer encryption using its decryption key K_n, and gets the original IP datagram. The exit AIN then sends the original IP datagram to the responder using the IP ESP+AH tunnel ($PVT_{n,r}$) between the exit AIN and the responder. In the meantime, AIN_n makes a connection between the $PVT_{n,r}$ and the $AIVT_{n-1,n}$.

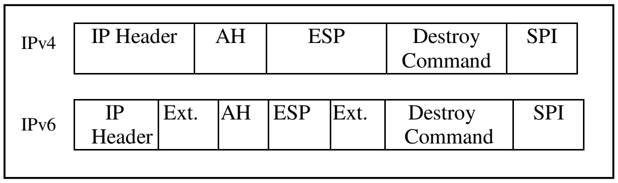
When a backward data stream is sent from the responder, an inverse processing is performed as the above, but the cryptographic operation is an encryption for each node except for the entrance AIN. The entrance AIN decrypts the backward data stream with K_2, then K_3,…, and final K_n.

4.5 Destroying AIVT Protocol

Every AIVT has a default inactive expiration time. It may also have another expiration time. An AIVT will be destroyed immediately after a destroying signal is sent, or when its expiration time or the default inactive expiration time is expired. A destroying AIVT signal can be made and sent by the initiator, responder and any AINs in the AIVT. There are several situations for destroying an AIVT as follows.

The first situation is for the expiration time and default inactive expiration time. If the AIVT has an expiration time and it has expired, all AINs automatically destroy the AIVT according to the expiration time. If the AIVT doesn't have an expiration time, all AINs also automatically destroy the AIVT according to its default inactive expiration time.

The second situation is for the initiator or responder sending a destroying signal. If either the initiator or responder sends a destroying signal to the entrance AIN or exit AIN for any reason (e.g. a session ends), the entrance AIN or exit AIN will create a destroying AIVT signal using the PVTs, send it to the next AIN, and destroy the AIVT. The next AIN finds its next SPI, and then creates a new destroying AIVT signal and sends it to its next AIN, and so on. Figure 5 depicts the AIVT destroying signal.



IPv4	IP Header	AH	ESP	Destroy Command	SPI

IPv6	IP Header	Ext.	AH	ESP	Ext.	Destroy Command	SPI

Figure 5. AIVT Destroying Signal.

The final situation is for any AINs sending a destroying AIVT signal. If any AIN sends a destroying AIVT signal for some reasons (e.g. it cannot connect the next AIN), the AIN will create two destroying AIVT signals, send them to the next AIN along the two directions using the PVTs, and destroy the AIVT. The destroying AIVT signal is the same as Figure 5.

5. IMPLEMENTATION

The easiest way to build the anonymous Internet without requiring the complete redesign and deployment of new client and server software is to make use of existing IPSec software technologies.

This section presents the interface specification between the components in the anonymous Internet. In order to provide some structure to this specification, we discuss the components in the order that data would move from the initiator to the responder in this section.

5.1 Entrance AINs

The interface between the initiator and the entrance AIN is defined as a standard IP ESP+AH tunnel, and is independent of any special applications. The initiator sends any IP datagrams to the entrance AIN using the IP ESP+AH tunnel. Figure 6 depicts the IP ESP+AH tunnel format.

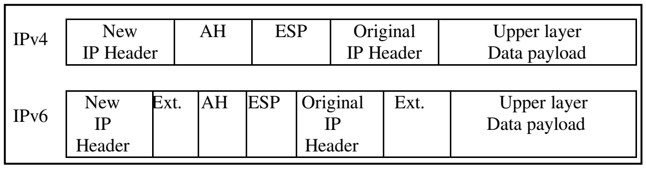


Figure 6. IP ESP+AH Tunnel between the Initiator and the Entrance AIN.

The initiator and the entrance AIN have four SAs for the bi-directional IP ESP+AH tunnel, respectively. Two of them support the forward data stream for IP ESP and IP AH, respectively. Another two support the backward data stream. The IP ESP+AH tunnel is built by the IKE protocol. Of course, they also can only use IP ESP or AH as the IP tunnel, but it would weaken the security.

The interfaces between the entrance AIN and next AIN include the PVT interface and the TVT interface. The PVT interface is defined as a standard IP ESP+AH tunnel. It is a long-term tunnel that can be built by the IKE protocol previously. Figure 7 depicts the PVT format for supporting the signaling transmission services.

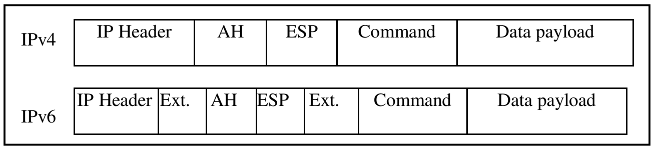


Figure 7. PVT Signaling Format.

The Command and Data payload fields are always encrypted using the IP ESP. The command is CREATE (0), DESTROY (1), or other. If the command is CREATE, the data payload should be *SPI+Creating Signal*. If the command is DESTROY, the data payload should be *SPI+Padding*.

The TVT interface is defined as a standard IP AH tunnel. It is a temporary tunnel that is built by the creating AIVT protocol. Figure 8 depicts the TVT format for supporting the data transmission services.

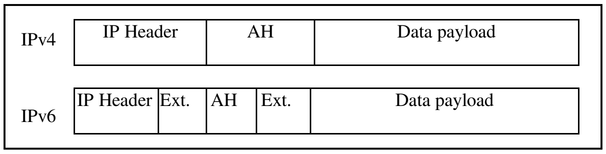


Figure 8. TVT Data Transmission Format.

The above data payload is encrypted with a nested encryption operation. Figure 9 depicts the nested encryption data through the TVT. For the forward data stream, the data payload is created by the entrance AIN using K_n, K_{n-1}, …, K_2 to encrypt the original IP datagram. For the backward data stream, the data payload is created by the exit AIN and intermediate AINs using their keys to encrypt the original IP datagram.

$$\boxed{E_{K_2}(\ E_{K_3}(\ldots E_{K_{n-1}}(\ E_{K_n}(\text{Original IP Datagram}))\ldots)}$$

Figure 9. Nested Encryption for Forward Data.

At the entrance AIN, a routing table is built for the AIVT according to the real IP source address, the original IP header of the initiator's IP datagram and the TVT. Two route messages are added to the routing table when the AIVT is built. Thus, the IP datagrams from the initiator to the responder are sent to the $TVT_{1,2}$, and the IP datagrams from the $TVT_{1,2}$ are sent to the initiator. Table 1 depicts the routing table in the entrance AIN.

Table 1. Routing Table in the Entrance AIN.

Source	Destination
$(IP_I, IP_R)_{Tunnel}$	$TVT_{1,2}$
$TVT_{1,2}$	$(IP_R, IP_I)_{Tunnel}$
…	…

Where $(IP_I, IP_R)_{Tunnel}$ means an IP ESP+AH tunnel with the original IP header from the initiator to the entrance AIN. The IP_I is the source IP address in the original IP header and may be real or unreal, and the IP_R is the responder's IP address.

5.2 Intermediate AINs

The interfaces between two neighboring intermediate AINs are the same as the interfaces between the entrance AIN and the next AIN, i.e. two interfaces: one for the PVT, and another for the TVT.

In an intermediate AIN, a routing table is built for the AIVT according to the its previous TVT and its next TVT in the AIN. Two route messages are added to the routing table when the AIVT is built. For example in the AIN_2, the IP datagrams are sent from the $TVT_{1,2}$ to the $TVT_{2,3}$ for the forward IP datagrams, and from the $TVT_{2,3}$ to the $TVT_{1,2}$ for the backward IP datagrams. Table 2 depicts the routing table in the AIN_2.

Table 2. Routing Table in the AIN_2 (Intermediate).

Source	Destination
$TVT_{1,2}$	$TVT_{2,3}$
$TVT_{2,3}$	$TVT_{1,2}$
…	…

In addition, each intermediate AIN only decrypts one layer of the forward data stream and encrypts one layer of the backward data stream, and sends it to the next AIN by the routing table.

5.3 Exit AINs

The interfaces between the next-to-last AIN and the exit AIN also are the same as the interfaces between the entrance AIN and the next AIN, i.e. two interfaces: one for the PVT, and another for TVT.

IP ESP+AH also is the desired interface between the exit AIN and the responder. The routing table is similar to the routing table of the entrance AIN. But the exit AIN is not sure whether the source IP address of the original IP datagrams is real or not. Table 3 depicts the routing messages added to the routing table it the exit AIN.

Table 3. Routing Table in the Exit AIN.

Source	Destination
$TVT_{n-1,n}$	$(IP_I, IP_R)_{Tunnel}$
$(IP_R, IP_I)_{Tunnel}$	$TVT_{n-1,n}$
…	…

Final, the exit AIN decrypts the last layer of the encryption data using its decryption key K_n for the forward data stream and sends it to the responder. For the backward data stream, the exit AIN encrypts the original IP datagram using K_n and sends it to the AIN_{n-1}.

5.4 Other Consideration

In order to provide ideal personal data protection against traffic analysis, the following techniques need to be included in our system.

- ***Packet Padding:*** In order to protect against the packet size correlation attack, the packet padding technique must be used in our system to make each packet transmitted between any two nodes (including the initiator and responder) have the same size.
- ***Dummy Traffic:*** In order to protect against the packet timing correlation attack, the dummy traffic technique must be used to send a constant number of packets per unit time.
- ***Ticket:*** In order to protect against the flooding attack, the ticket technique can be added to our system, i.e. each user must show that he/she is allowed to use the system at the respective time slice by providing a ticket valid for the certain slice, only.

Final, in order to be effective for our system, every entrance and exit AIN also take part in the role of the intermediate AINs.

6. VULNERABILITIES

Our Anonymous Internet Communication provides personal data protection against traffic analysis at IP layer. It can prevent the following attacks:

- ***Collusion Attack:*** It can prevent the collusion attack among any intermediate AINs and outside eavesdroppers because every intermediate AIN only knows its previous and next AINs. But it does not provide the protection against the attacks from the entrance AIN.
- ***Message Coding Attack:*** It provides the protection against the message coding attack using the nested encryption.
- ***Message Volume Attack:*** It can prevent the message volume attack using the packet padding technique.
- ***Timing Attack:*** It can prevent the timing attack using the dummy traffic techniques [2].
- ***Flooding Attack:*** It can prevent the flooding attack using the "ticket" technique [2].
- ***Replay attack:*** It provides the protection against the replay attack using IP AH, and data origin authentication and integrity protection. It also provides the data confidentiality protection with IP ESP for the signaling data and with the nested encryption for the transmission data.

In our system, the entrance AIN retains all information necessary for the anonymous connection such as the initiator, the responder and all AINs. Thus, the entrance AIN must be a trusted node by the initiator. In addition, any outside eavesdroppers between the AINs would not be able to discover anything about the data transfer.

7. CONCLUSION

IPSec is an excellent approach for secure communication over the Internet. Based on IPSec, we present an anonymous Internet communication using IP tunnel at IP layer. It provides bi-directional, real-time communication for any TCP and UDP applications, including the end-to-end IPSec, VPN, SSL, etc.

In this paper, however we don't discuss the network information database, routing and how to learn the topology of the anonymous Internet. This will be the subject of a future paper.

Because of its importance to the acceptance of e-business systems, we expect that more and more people will be involved in the research and development of an anonymous Internet including pseudonym IP. All of these efforts will make our Internet more robust and secure.

ACKNOWLEDGMENTS

We would like to thank all members of IIT at the NRC of Canada for their support towards our R&D projects in Privacy Protection. We also thank our European Community partners in the Privacy Incorporated Software Agent (PISA), an IST-EU Fifth Framework Project.

REFERENCES

[1] D.Chaum. Untraceable Electronic Mail, Return Address, and Digital Pseudonyms. Communications of the ACM, Vol.24, No.2, pages 84-88, 1981.

[2] O.Berthold, H.Federrath and S.Kopsell. Web MIXes: A System for Anonymous and Unobservable Internet Access. In H.Federrath, editor, Anonymity 2000, Volume 2009 of Lecture Notes in Computer Science, pages 115-129, Springer-Verlag, 2000.

[3] D.Goldschlag, M.Reed and P.Syverson. Hiding Routing Information. In R.Anderson, editor, Information Hiding: First International Workshop, Volume 1174 of Lecture Notes in Computer Science, pages 137-150, Springer-Verlag, 1996.

[4] M.Reed, P.Syverson and D.Goldschlag. Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications, Vol.16, No.4, pages 482-494, May 1998.

[5] M.Reiter and A.Rubin. Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, Vol.1, pages 66-92, 1998.

[6] P.Boucher, A.Shostack and I.Goldberg. Freedom Systems 2.0 Archtecture. December 2000. Available at http://www.freedom.net/info/whitepapers /Freedom_System_2_Architecture.pdf.

[7] O.Berthold, H.Federrath and M.Kohntopp. Project "Anonymity and Unobservability" in the Internet. Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions, Toronto, ON Canada, pages 57-68, April 2000.

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

[9] A.O.Freier, P.Karlton and P.C.Kocher. The SSL Protocol: Version 3.0. Internet Draft, Netscape Communications, March 1996. Available at http://home.netscape.com/eng/ssl3/ssl-toc.html.

[10] S.Kent and R.Atkinson. IP Authentication Header. IETF RFC 2402, November 1998.

[11] S.Kent and R.Atkinson. IP Encapsulating Security Payload (ESP). IETF RFC 2406, November 1998.

[12] W.Dai. Pipenet 1.1. 2000. Available at http://www.eskimo.com /~weidai/pipenet.txt.

Biography

Ronggong Song received his B.Sc degree in mathematics in 1992, M.Eng degree in computer science in 1996, Ph.D. in network security from Beijing University of Posts and Telecommunications in 1999. He had employed as Network Planning Engineer at Telecommunication Planning Research Institute of MII, P.R.China, and Postdoctoral Fellow at University of Ottawa, Canada. Now, he is working at NRC of Canada. His research interests are privacy protection, network security, e-commerce, IP mobility.

Larry Korba is the group leader of the Network Computing Group of the National Research Council of Canada in the Institute for Information Technology. He is currently involved in several projects related to security and privacy. His research interests include privacy protection, network security, and computer supported collaborative work.