PBS: Signaling Architecture for Network Traffic Authorization

Se Gi Hong
Columbia University, New York, NY
segihong@cs.columbia.edu

Henning Schulzrinne
Columbia University, New York, NY
hgs@cs.columbia.edu

Swen Weiland
University of Göttingen, Germany
weiland@informatik.uni-goettingen.de

***Abstract*—We present a signaling architecture for network traffic authorization, Permission-Based Sending (PBS). This architecture aims to prevent Denial-of-Service (DoS) attacks and other forms of unauthorized traffic. Towards this goal, PBS takes a hybrid approach: a proactive approach of explicit permissions and a reactive approach of monitoring and countering attacks. On-path signaling is used to configure the permission state stored in routers for a data flow. The signaling approach enables easy installation and management of the permission state, and its use of soft-state improves robustness of the system. For secure permission state setup, PBS provides security for signaling in two ways: signaling messages are encrypted end-to-end using public key encryption and TLS provides hop-by-hop encryption of signaling paths. In addition, PBS uses IPsec for data packet authentication. Our analysis and performance evaluation show that PBS is an effective and scalable solution for preventing various kinds of attack scenarios, including Byzantine attacks.**

I. Introduction

Currently, the Internet architecture allows any node to inject IP packets into the network without requiring explicit permission from the intended receiver. As Internet usage and applications have exploded over the past decade, this simple architecture has enabled misuse of the network itself. Indeed, it has made Denial-of-Service (DoS) attacks possible and often lead to degraded quality of service. DoS attacks have become a particular problem since so-called botnets have made it possible for remote attackers to commandeer end systems and use them to disrupt communication.

A report on network security threats published by Symantec [1] estimates an average of 6,110 DoS attacks per day during the first six months of 2006. The report also observed an increase of the number of active botnet computers per day from 10,352 in 2005 to 52,771 in 2007. Arbor networks [2] provided a bandwidth measurement of DoS attacks indicating a steady growth. The largest attack size in 2007 exceeded 40 gigabits per second.

Existing proposals on how to prevent DoS attacks can be classified in two ways. A reactive approach monitors network traffic. If an attack is detected, an action is taken against it. Filtering-based approaches [3][4][5] fall into this category. A proactive approach sets up a rule that all data flows between a sender and a receiver have to follow. Data flows that violate the rule are simply dropped. SOS [6] and capability-based approaches [7][8][9][10] are examples of proactive approaches.

Each approach has advantages and disadvantages. An advantage of a reactive approach is its ability to adapt its counter-attack strategy dynamically as it monitors on-going attacks. However, there are two disadvantages in a reactive approach. It is not always possible to differentiate legitimate packets from malicious packets [6], and by the time a reactive algorithm detected and acted against an attack, the attacker might have already accomplished his objective. In contrast, a proactive approach can prevent such an attack from taking place preemptively by setting up appropriate system rules. However, such an approach bears a risk of letting an attacker into the system if the attacker is able to circumvent the rules. Neither reactive nor proactive approach is effective against a compromised router.

We developed a new approach to prevent unauthorized traffic. Our approach, called Permission-Based Sending (PBS), is a hybrid of the proactive and reactive approaches, combining the benefits of two approaches and mitigating their disadvantages at the same time. We use an explicit permission to give legitimate packets the authority to send (proactive approach), and use a monitoring mechanism to detect and react against attacks (reactive approach). The explicit permission allows differentiation of benign traffic from malicious one, and can limit the severity of attacks. The monitoring mechanism can provide a second line of defense against malicious traffic, which may have circumvented the permission-based mechanism.

To implement our hybrid approach, we developed a secure and robust signaling architecture. In this architecture, the end hosts send signaling messages along the path of data flow in order to install permission states into the routers in the path. This is called *on-path* signaling. The signaling messages also contain information about traffic volume that can be used for network monitoring. To securely set up the permission state, signaling messages are protected end-to-end against alternation using digital signature. The channel security (TLS and DTLS) is used hop-by-hop for integrity and confidentiality of signaling messages. The soft-state mechanism of PBS supports the robustness against the state changes. Data packets use IPsec to authenticate the origin.

We analyze various attack models and show how PBS can be used to counter those attacks. In particular, our PBS can effectively prevent Byzantine attacks, which have been considered difficult to counteract. Furthermore, our evaluation shows that the signaling overhead is small enough to make PBS a practical solution in large scale networks.

The remainder of this paper is organized as follows. In Section II, we give an overview of PBS. We present security analysis in Section III. In Section IV, we provide the architecture and implementation details. We present performance evaluation in Section V. In Section VI, we discuss deployment issues of PBS. Finally, we discuss related work in Section VII.

II. PBS OVERVIEW

The goal of permission-based PBS is to allow the legitimate senders to send data by granting permission and drop the unauthorized packets by default.

A. Design goal

There are five design requirements for PBS: it must be deployable, distributed, robust, secure, and scalable.

PBS should work in the current Internet infrastructure. For example, it should not modify IP packet headers or TCP/UDP packet headers. Thus, PBS uses signaling messages to set up and manage permission state, instead of piggybacking the permission information in the IP packet header or TCP/UDP packet header. PBS uses existing security protocols, such as IPsec, TLS and digital signatures.

PBS should be a distributed system to eliminate the necessity of managing a central server. Thus, the permission state is managed between the receiver and the sender along the data path by signaling. A subset of routers keeps state for a data flow, and monitors whether the flow is authorized.

PBS should be robust in the face of changes, such as routing and permission. Soft-state of PBS supports the robustness of the system. Thus, the permission state is periodically refreshed by signaling messages. At the absence of the refresh of a state, the permission state is eliminated.

The permission setup and management should be secure. Therefore, the signaling messages that install and modify the permission state and distribute cryptography keys are protected by cryptography algorithms. Data packet is also protected by against alternation.

PBS should be scalable to be applicable in large scale networks. In PBS, PBS functionality does not need to be implemented in all routers. Thus, some of the routers that have PBS functionality properly handle the authorization of data flows. In addition, the computational and signaling overhead is small for scalability as we show in Section V.

B. Explicit permission using signaling

For permission state setup and management, PBS uses a suite of IP signaling protocols that have been developed by the IETF Next Steps in Signaling (NSIS) working group [11]. The NSIS protocol suite consists of two protocol layers: the NSIS Transport Layer Protocol (NTLP) and NSIS Signaling Layer Protocols (NSLPs) [12].

The General Internet Signaling Transport (GIST) [13] implements NTLP. The main purposes of GIST are to determine how to reach the next node along the data path (routing) and deliver signaling messages to the peer (transport). GIST supports some design requires of PBS, such as on-path signaling, robustness against route changes, ability to work on the current networking architectures, and scalability. GIST provides on-path signaling by using underlying routing state information to deliver signaling messages along the data path. GIST is robust to route changes because it detects the route change and informs the NSLP layer about the changes (see [13] for more details). GIST reuses the existing transport layer protocols and security protocols, so that it does not require modification of current network protocols. As Fu et al. [14] evaluated the performance of the GIST protocol, GIST supports scalability. The authors observed that "A GIST node serving 45,000 signaling sessions is found to consume small amount of CPU and memory" [14]. Not all routers need to have NSIS functionality. The signaling messages bypass the router, which does not have the functionality.

On top of GIST, PBS needs a way to authorize network traffic, so we developed a new NSLP, the PBS NSLP [15]. The NTLP (GIST) handles all incoming signaling messages and it passes the PBS-related signaling messages to the PBS NSLP layer. There are two message types in the PBS NSLP; namely Query (Q) and Permission (P) messages. The Query message is sent by a sender to request permission to send data. It contains the flow identification object, whose information is 5-tuple (source IP address, destination IP address, source port, destination port, and protocol identifier), describing data flow. It also contains a requested volume of data in bytes for a flow. The Permission message is sent by the receiver who grants the permission to the sender along the reverse path of the Query message. The reverse path is set up by the GIST reverse routing state. The Permission message is used to set up (grant), remove (revoke) and modify permission state for a flow. The Permission message contains the flow identification, the allowed volume in bytes, time limit for the permission, and the refresh time for soft-state. The PBS nodes, which are routers and end hosts that have PBS functionality, store these information to keep track of permission states. The delivery of signaling messages is performed hop-by-hop approach between the adjacent PBS nodes. In other words, each PBS node forwards the signaling message to the next PBS node. The Query and Permission messages are periodically transmitted to establish soft-state that enables the detection of permission state and security algorithm changes.

PBS supports the asymmetric transmission of Query and Permission messages. After the permission state is set up, the Permission message can be sent when a receiver wants to change (revoke and modify) the permission state and security mechanism without receiving a Query message.

C. Security of messages

Many forms of DDoS attacks employed today do not need to spoof source addresses. However, IP address spoofing is still prevalent [16] [17]. "Approximately 25% of netblock and ASes allow some form of spoofing," according to Beverly and Bauer [16]. Therefore, authentication and integrity of signaling messages are required for secure permission state setup and management.

PBS uses a public key cryptography mechanism for the authentication and integrity of signaling messages. Each sender and receiver generates a public/private key pair, and generates a digital signature by encrypting the objects of signaling messages using its own private key (i.e., the sender encrypts the objects of the Query message with its private key and the receiver encrypts the objects of the Permission message with its private key). Each public key in the form of the X.509 certificate, which is certified by a trusted third party (certificate authority), is distributed by a signaling message to the PBS nodes. The Query message carries the sender's public key and the Permission message carries the receiver's public key. To validate the authentication and integrity of the signaling messages, each PBS node decrypts the digital signature using the distributed public key.

For the authentication and integrity of data packets, IPsec Authentication Header (AH) is used. The Permission message carries the shared key and security parameter index (SPI), which are generated by the receiver and will be used for IPsec. When each PBS node receives the Permission message, it stores the shared key and installs the security association (SA). For each flow, the SA has field values for destination IP address, IPsec protocol (AH or ESP) and SPI. To securely deliver the key and SPI value, channel security protocol (TLS or DTLS) is used between adjacent PBS nodes. PBS functionality allows PBS routers to validate the IPsec that uses transport mode between the two end hosts (sender and receiver).

For the authentication data field in IPsec AH, the sender uses symmetric key cryptography or public key cryptography. In symmetric key cryptography, the shared symmetric key that is delivered in the Permission message is used for the encryption. The public key cryptography method entails using the sender's private key for encryption. The receiver has the right to choose a cryptography algorithm for IPsec based on the policy, network and applications, and this notification is carried in the Permission message.

Fig. 1 shows the secure two-way handshakes for permission state setup and how PBS can prevent attack flows. Since the attacker does not have the shared key, the attack flow failed during IPsec verification. Thus, it is dropped at the first router ($R1$).

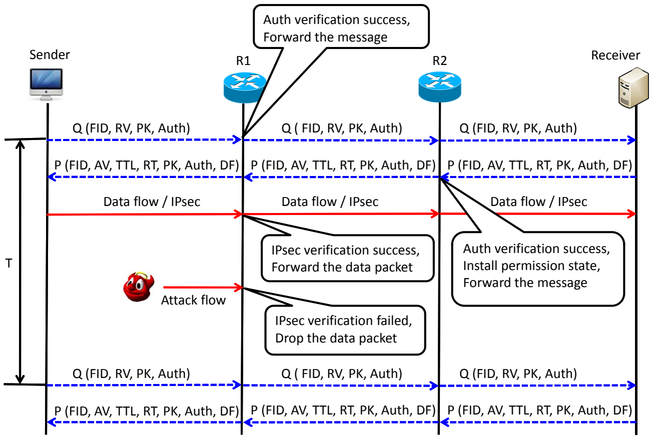

Auth verification success,
Forward the message
Sender
R1
R2
Receiver
Q (FID, RV, PK, Auth)
Q (FID, RV, PK, Auth)
Q (FID, RV, PK, Auth)
P (FID, AV, TTL, RT, PK, Auth, DF)
P (FID, AV, TTL, RT, PK, Auth, DF)
P (FID, AV, TTL, RT, PK, Auth, DF)
Data flow / IPsec
Data flow / IPsec
Data flow / IPsec
T
IPsec verification success,
Forward the data packet
Auth verification success,
Install permission state,
Forward the message
Attack flow
IPsec verification failed,
Drop the data packet
Q (FID, RV, PK, Auth)
Q (FID, RV, PK, Auth)
Q (FID, RV, PK, Auth)
P (FID, AV, TTL, RT, PK, Auth, DF)
P (FID, AV, TTL, RT, PK, Auth, DF)
P (FID, AV, TTL, RT, PK, Auth, DF)

Fig. 1. PBS basic operation —— FID: Flow identification; RV: The volume of data that the sender requests; AV: The volume of data that the receiver grants; TTL: Time limit for the permission; RT: refresh time for soft-state; PK: The certificate of a public key; Auth: The authentication field (digital signature) that is encrypted by one's private key; DF: Defense object. It has solution field (the indicated solution against the attack), IPsec AuthAlgo field (the cryptography algorithm for the IPsec authentication field), a shared key, and SPI value.

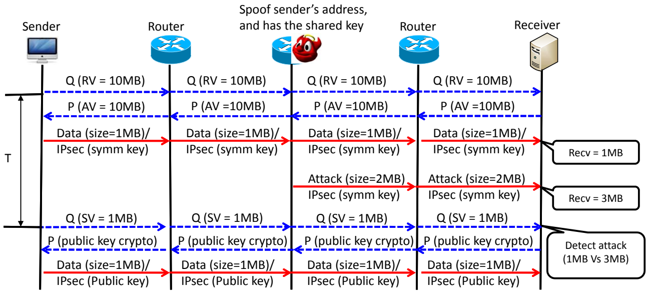


Fig. 2. Basic operation of the PBS Detection Algorithm (SV: The volume of data that the sender has sent; Recv: The volume of data that the receiver has received; size: data size)

D. *Monitoring and reaction against DoS attacks*

Other routers that do not have PBS functionality cannot generate bogus data packets because they do not have the shared key. A compromised PBS router that knows the shared key, however, can generate and insert attack packets when symmetric key cryptography is used in IPsec AH. Furthermore, an off-path attacker (i.e., external attacker) might obtain the shared key by controlling compromised PBS routers. Compromised routers, which may or may not be PBS routers, can drop legitimate packets. To prevent the attacks in this Byzantine network, PBS requires monitoring of network traffic and detecting attacks. The detection algorithm is called PBS Detection Algorithm (PDA). PDA uses existing signaling messages (Query and Permission messages) and soft-state of the system. A sender periodically sends the Query message that contains a volume of data that it has sent after the permission is granted. The attacker cannot modify the volume information in the Query message since the Query message is protected by public key cryptography. The receiver compares the volume of data in the signaling message with the volume of data that has been received. If both of the volumes are the same, there is no attack, but if both of them are different, the receiver suspects that there is an attack. Based on the detection, a receiver requests the senders to react against the attack.

Fig. 2 shows the basic operation of PDA. We assume that the receiver grants permission to the sender to send a flow of size 10 MB. After setting up the permission state, the sender sends data packets whose total volume is 1 MB. Since a compromised router has the shared key, it can generate attack packets with the correct IPsec header. It sends attack packets whose total volume is 2 MB. After period T, the sender sends a Query message that contains a volume (1 MB) of data that it has sent. The receiver can detect the attack by comparing the volume (1 MB) in the Query message and total volume of data (3 MB) that it has received. After the receiver detects the attack, it sends the Permission messages with an indication to use public key cryptography to generate authentication field of IPsec header. Therefore, after the sender uses public key cryptography for the IPsec, the attack packets are dropped at a router because of the IPsec verification process.

PDA can detect the packet dropping attacks by a compromised router. A compromised router drops all packets (including signaling messages) or selected packets (e.g., every n packets). When a compromised router drops all packets, since the sender does not receive a Permission message, the sender suspects that the packets have been dropped. Therefore, it changes the path. When a compromised router drops some data packets, the amount of volume that the receiver has received and the volume information in the Query message differ, so the receiver suspects that packets have been dropped and sends a Permission message indicating a request to change path. To change the path, the sender can use a relay node used for tunneling or path diversity by multihoming. The method for path changes is out of scope of this project.

Data packet loss due to natural causes is also possible, and this is not an attack. Because of PDA, the natural packet loss might be regarded as a dropping attack. To avoid this, we apply a threshold-based decision scheme. If the difference between the amount of delivered packets and the volume information in the Query message is within a defined threshold, this is not regarded as a dropping attack. However, if the difference is bigger than the threshold, this is regarded as a dropping attack. The threshold value can be defined by the receiver based on the network environment. PDA can also detect the heavy congestion link where there is significant packet loss, and it triggers the path changes. Because of the retransmission scheme in NTLP (GIST) for the signaling message, we can say that the signaling message delivery is reliable. Thus, if there is signaling message loss, this means that there is a dropping attack or the link is heavily congested. Thus, the sender changes the path to avoid signaling message dropping.

E. Robustness against the route change

Route changes, which occur by router failure, link failure, or router restart, can affect the performance of PBS since the new router that is on the new path is not aware of the permission state of the flows. In PBS, however, the soft-state of GIST can detect the route change and inform the PBS NSLP about the changes, so the flow session can be set up at the new router. Furthermore, the soft-state of PBS NSLP messages is used for the detection of route changes. The new router, which gets the new PBS Query and Permission message, updates the permission state. The old router, which does not get signaling messages for the flow after a soft-state period, removes the permission state of the flow. The state updating time for the route changes depends on the soft-state period. Before the new permission state is installed on the new router, the flows are rate-limited and volume-limited at the new router.

III. Security Analysis of PBS

A. Attack handling in PBS

1) Trustworthy networks: Since the core network is trustworthy, routers are not compromised. Thus, there is no on-path attack. However, an off-path attacker (external attacker) may insert bogus packets into the data path.

When an off-path attacker does not spoof the address, the attack packets can be dropped at a router by checking the 5-tuple of the packets since they do not have permission.

If the attacker spoofs the address of one of the legitimate senders who has permission, the legitimate sender cannot send packets or can send packets with only a small portion of its permission. However, even if the attacker is in the same subnet as the legitimate sender and spoofs the sender's address, the attack packets will not pass IPsec validation and will be dropped.. Since the network is trustworthy, symmetric key cryptography algorithm for IPsec is a good solution for this attack.

2) Byzantine networks: In Byzantine networks, in which we trust neither the sender nor the routers, the off-path attacks can be prevented by IPsec, similar to the method used in trustworthy networks.

Unlike trustworthy networks, in Byzantine networks, the compromised PBS router can inject the attack packets and drop the legitimate packets. However, PDA can detect the packet injection and dropping as shown in Section II-D. Since the compromised router has a session key, IPsec using symmetric key cryptography cannot prevent the attack. Thus, public key cryptography should be applied for the authentication field of IPsec AH. To avoid a compromised router that drops legitimate packets, changing the data flow path is needed.

PDA can detect all attacks except for a replacement attack because PDA is based on comparing the volume of data. In the packet replacement attack, the attacker does not generate or drop the packets, but changes the content of the packet. Thus, this attack cannot be detected by PDA. One solution to prevent the attack is to use message security by public key cryptography for IPsec and apply to every data packets. However, it requires computational overhead. Therefore, this message security should be applied minimally. However, if the network system requires high-end security, such as a military system, the message security to every message is required even though the system has to pay more cost.

B. Security issues

1) Permission granting process: The permission granting process depends on the policy of the receiver. In PBS, we assume that the receiver has a white list and a black list. The sender in a white list can get the permission, but the sender in a black list cannot get the permission from the receiver. This white and black listing, however, has an introduction problem. The introduction problem is on deciding whether the receiver gives the permission to the sender, who is not on either of the lists. In the current PBS system, we assume that anonymous

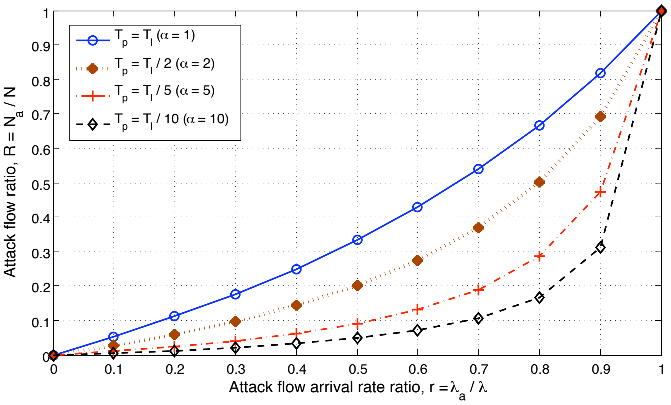


Fig. 3. Attack flow ratio

user will get limited permission until the receiver suspects that the user is not an attacker.

2) Denial-of-Permission: An attacker can send a lot of Query messages, so the PBS router needs to validate the Query messages and this requires computational overhead. We call this attack a denial-of-permission attack. To prevent this attack, we assume that computational puzzles [18][19] are used for rate-limiting the Query message in the PBS system.

These security issues are the future work of this project.

C. Detection delay and number of attack flows

Since some of the attack flows can pass the router and reach the receiver before the attack flows are detected by PDA, we need to decrease the number of attack flows. As the attack flows are detected quickly, the number of attack flows decreases. Detection delay of attack flows by PDA depends on the soft-state period of the signaling messages. We analyze how the soft-state period affects the number of attack flows, and how PDA can reduce the number of attack flows.

We assume that the attack flow arrival rate at the receiver, λ_a, and the legitimate flow arrival rate, λ_l, follow Poisson distribution. Thus, the total flow arrival rate at the receiver, λ, is the sum of the two arrival rates. From the Little's law, we can find the average number of legitimate flows and attack flows. The expected lifetime of all the flows, $E[T_L]$, is

$$E[T_L] = \frac{\lambda_a}{\lambda} E[T_a] + \frac{\lambda_l}{\lambda} E[T_l]$$

where $E[T_a]$ is the expected lifetime of attack flows and $E[T_l]$ is the expected lifetime of legitimate flows. We assume that the attack flows continue until they are detected and terminated by PDA. Thus, the lifetime of attack flows in the system is equal to the detection delay. Let the soft-state period of signaling message, T_P, be T_l/α where α is equal to or larger than one. Since the arrival of attack flow between the soft-state period follows uniform distribution, the average delay of attack detection is $T_P/2$. Therefore, the attack flow ratio, R, is

$$R \;=\; \frac{E[N_a]}{E[N]} = \frac{\lambda_a E[T_a]}{\lambda E[T_L]} = \frac{r}{r + (1-r)2\alpha}$$

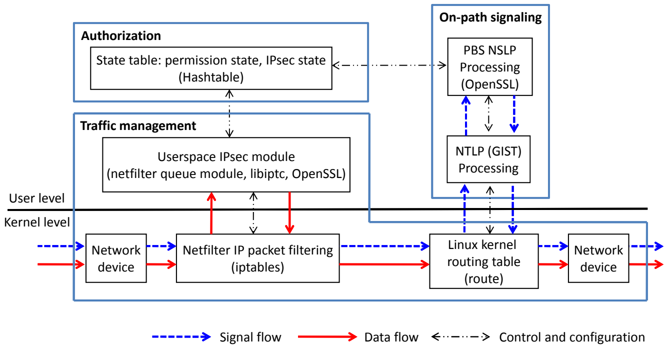


Fig. 4. PBS implementation

where r is the ratio of attack flow arrival rate over total flow arrival rate ($r = \lambda_a/\lambda$).

Figure 3 shows the attack flow ratio, R, with various r and T_p. In the figure, even though attack flow arrival rate ratio is 0.8 (i.e., attack flow arrival rate is much larger than legitimate flow arrival rate), the attack flow ratio is less than 0.2 when T_p is $T_l/10$. This result shows the small soft-state period decreases the detection delay, and because of the detection, the number of attack flows is reduced. There is, however, a trade-off between detection delay and signaling overhead (message and processing overhead). As the period decreases, the signaling overhead increases.

IV. PBS Architecture and Implementation

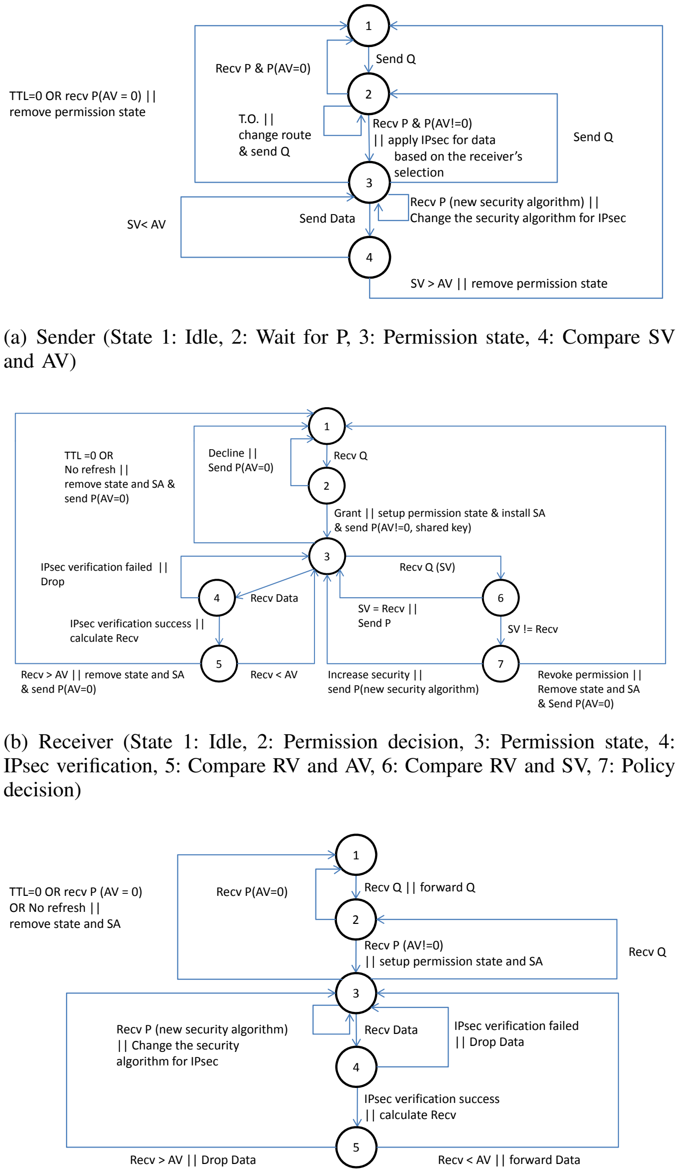
PBS has three components: on-path (path-coupled) signaling, authorization, and traffic management. We have implemented the components of PBS. Fig. 4 shows our PBS implementation architecture.

A. On-path signaling

We are building PBS NSLP on the GIST implementation [20], which provides the channel reliability (C-mode) and security (TLS). PBS NSLP states are managed by a finite state machine (FSM). Fig. 5 shows the FSM of PBS NSLP. PBS NSLP parses and creates signaling messages at each node. OpenSSL [21] is used for implementing cryptography algorithms and processing X.509 certificates. The communication between GIST and PBS NSLP is performed by Unix sockets. More details about the signaling message formats and PBS NSLP specification can be found in the Internet draft [15].

B. Authorization

The authorization component decides whether to grant permission (amount of volume) and time limit for a flow. Another main objective of this component is to detect and identify attacks. The receiver's policy for prevention mechanism, such as selection of cryptography algorithm, against the detected attack is located in this component. Authorization manages the permission state table, which is implemented by a hashtable, for each flow. Fig. 6 shows the permission state table.


1
Send Q
Recv P & P(AV=0)
2
TTL=0 OR recv P(AV = 0) || remove permission state
T.O. || change route & send Q
Recv P & P(AV!=0) || apply IPsec for data based on the receiver's selection
Send Q
3
Recv P (new security algorithm) || Change the security algorithm for IPsec
SV< AV
Send Data
4
SV > AV || remove permission state
1
TTL =0 OR No refresh || remove state and SA & send P(AV=0)
Decline || Send P(AV=0)
Recv Q
2
Grant || setup permission state & install SA & send P(AV!=0, shared key)
3
IPsec verification failed || Drop
Recv Q (SV)
4
Recv Data
6
SV = Recv || Send P
SV != Recv
IPsec verification success || calculate Recv
5
7
Recv > AV || remove state and SA & send P(AV=0)
Recv < AV
Increase security || send P(new security algorithm)
Revoke permission || Remove state and SA & Send P(AV=0)
1
TTL=0 OR recv P (AV = 0) OR No refresh || remove state and SA
Recv P(AV=0)
Recv Q || forward Q
2
Recv P (AV!=0) || setup permission state and SA
Recv Q
3
Recv P (new security algorithm) || Change the security algorithm for IPsec
Recv Data
IPsec verification failed || Drop Data
4
IPsec verification success || calculate Recv
5
Recv > AV || Drop Data
Recv < AV || forward Data

(a) Sender (State 1: Idle, 2: Wait for P, 3: Permission state, 4: Compare SV and AV)

(b) Receiver (State 1: Idle, 2: Permission decision, 3: Permission state, 4: IPsec verification, 5: Compare RV and AV, 6: Compare RV and SV, 7: Policy decision)

(c) Router (State 1: Idle, 2: Wait for P, 3: Permission state, 4: IPsec verification, 5: Compare RV and AV)

Fig. 5. Finite state machine (Event || Action) (Q: Query message, P: Permission message, T.O.: Time out, AV: The number of bytes that the receiver allows, SV: The number of bytes that the sender has sent, Recv: The number of bytes that the node has received)

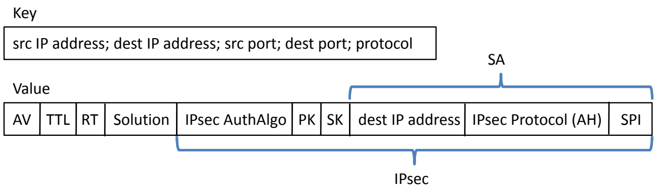


Fig. 6. Permission state table. The 5-tuple is the key in the hashtable. (AV: The number of bytes that the receiver grants; TTL: Time limit for the permission; RT: The refresh period; Solution: The indicated solution against the attack (change path, use public key cryptography, etc); PK: The sender's public key; SK: The shared key)

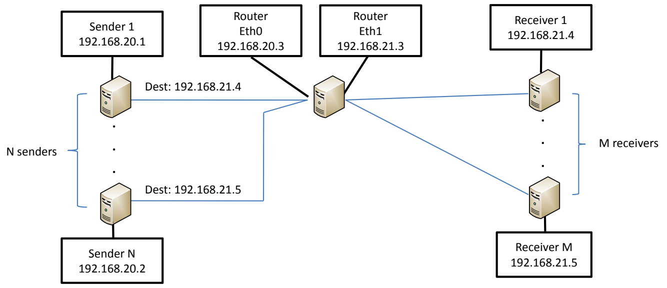


Fig. 7. Testbed. The router has two network interfaces (one interface is connected to senders' subnet, and the other interface is connected to receivers' subnet.

C. Traffic management

The traffic management handles all incoming packets, including signaling messages and data packets. It passes signaling messages up to the on-path signaling component. Based on the permission state of flows, the traffic manager screens the data packets to see whether the data packets are authorized. An IP packet filter is used to filter the unauthorized packets. IPsec header is verified in this component. To see whether the flow exceeds the given permission, this component monitors the volume of the data flow that it has received since the permission state was set up.

We implemented a userspace IPsec module which is a modular IPsec stack that relies on user space by using netfilter [22]. `libiptc` interfaces filter tables in the kernel space and `iptables` filters IP packets. `netfilter queue` module gets the packets to a user space if a rule matches. We set up Linux routing tables using `route`.

V. Performance evaluation

To show that PBS supports scalability, which is one of the design goals of PBS, we measure the signaling overhead, the CPU usage, memory overhead, and delays. Since the performance of IPsec is evaluated in [23], we focus on the performance evaluation of signaling messages. Fig. 7 shows the testbed setup for performance measurement. The machines are running Linux with kernel version 2.6.23, have AMD Opteron 2.2GHz CPUs, and 2GB RAM each.

A. Signaling overhead

The signaling overhead can be determined by the size of signaling message and frequency of the messages. There are two kinds of signaling messages: NTLP (GIST) and PBS NSLP messages. Fu et al. measured the GIST handshake signaling messages [14]. Thus, we focus on the PBS NSLP signaling messages. In PBS NSLP, Query and Permission messages are the NSLP data messages.

The total generated NSLP messages during a permission session, L_{nslp}, is

$$L_{nslp} = (L_Q + L_P) \times (\lfloor \frac{T_L}{T_P} \rfloor + 1)$$

where L_Q is the size of Query message, L_P is the size of Permission message, T_L is the lifetime of the session, and T_P is the soft-state period of NSLP.

Table I shows the message size of PBS NSLP signaling of NSIS layer. The size depends on the X.509 certificate size that the signaling messages carry and the symmetric key size that the Permission messages carry. We vary the public key algorithms for the authentication field of signaling message. The three public key algorithms have the same security level (80-bit security level). The signaling message overhead also depends on the protocol header sizes.

The total generated NSIS signaling message during a permission session, L_{signal}, is the sum of the total generated NSLP messages and GIST messages.

The signaling message overhead ratio, R_s, is

$$R_s = \frac{L_{signal}}{L + L_{signal}}$$

where L is the total data size including headers of link layer, network layer, and transport layer protocol. Thus, L is

$$\begin{aligned} L &= L_d + N_p \times L_{header} \\ N_p &= \lceil \frac{L_d}{MSS} \rceil \end{aligned}$$

where L_d is the size of data message, N_p is the number of packets for the original data message, L_{header} is the sum of the header size of all layer protocols, and MSS is the maximum segment size.

We assume that the permission state is set up for a flow of streaming video using UDP. The size of the flow is 4 GB and the running time of the flow is 90 minutes. Thus, the permission state lasts for 90 minutes. We assume that the soft-state periods of PBS NSLP and GIST are 60 seconds. Table I shows the bandwidth usage of the signaling messages during the permission session and signaling message overhead ratio.

TABLE I
PBS NSLP SIGNALING MESSAGE SIZE, AND BANDWIDTH AND SIGNALING OVERHEAD RATIO FOR 4GB VIDEO STREAMING WHOSE RUNNING TIME IS 90 MINUTES—— THE FIRST COLUMN OF THE TABLE REPRESENTS THE PUBLIC KEY CRYPTOGRAPHY FOR THE AUTHENTICATION FIELD OF THE SIGNALING MESSAGE.

Parameters	Query (bytes)	Permission (bytes)	Bandwidth usage (kbits/sec)	Overhead ratio
RSA-1024	1153	1189	0.376	0.000062
DSA-1024	1252	1292	0.403	0.000066
ECC-192	917	957	0.313	0.000051

B. Computational overhead

We measure the CPU usage to handle signaling messages at a router. Since the sender generates a few sessions at a time, the CPU usage of a sender is not a problem. When a receiver is an end user host, there will be only a few simultaneous sessions. Therefore, the CPU overhead is unlikely to become significant. When a receiver is a server that handle much traffic, some form of load balancing strategy is likely to be part of its deployment, and this will spread the CPU load caused by PBS. Thus, we mainly consider the router's CPU usage. Fig.8 shows the CPU usage of PBS signaling message handling. The CPU usages of PBS NSLP layer with different transport layer protocols and security protocols are the same since the signaling message delivery and channel security are performed at the GIST layer and the NSLP layer is for the signaling message verification and parsing. However, the CPU usage varies based on the transmission protocol and security protocol in the GIST layer. The CPU usage is the lowest when UDP is used and the highest when TLS is used. Even though TLS is used for the delivery of Query and Permission messages, the router can handle 600 Query and 600 Permission messages per second. It means that if the period of soft-state in PBS NSLP is 60 seconds (short stream flows whose lifetime is less than 60 seconds get one-time permission), the router can handle 36,000 sessions concurrently.

Lee et al. [24] measured the network flows at the edge routers of University of Auckland in 2006. They show the average flow departure rate and lifetime of flows. By using their data and Little's law and assuming that arrival rate and departure rate are same, we can get the average number of concurrent flows, which is 10,000 flows, and the maximum number of concurrent flows, which is 20,000 flows. Thus, PBS can be applied to the edge router. Since the computer that we used for our performance evaluation is old and slow, if we test the CPU usage at a faster CPU, the CPU usage can be lower and the router can handle more signaling messages.

C. Memory overhead

Since PBS is based on permission state, there is a memory usage overhead for storing permission state (including keys) for each flow. The memory size for storing session keys depends on the key size and concurrent number of sessions. If there are 10,000 concurrent sessions, then session key storage requires 0.2 MB for HMAC-SHA1 and 1.28 MB for RSA-1024. If we assume that each flow requires 100 bytes for storing other information, it only requires 1MB with 10,000 concurrent sessions. Therefore, the total memory usage for PBS is not a problem.

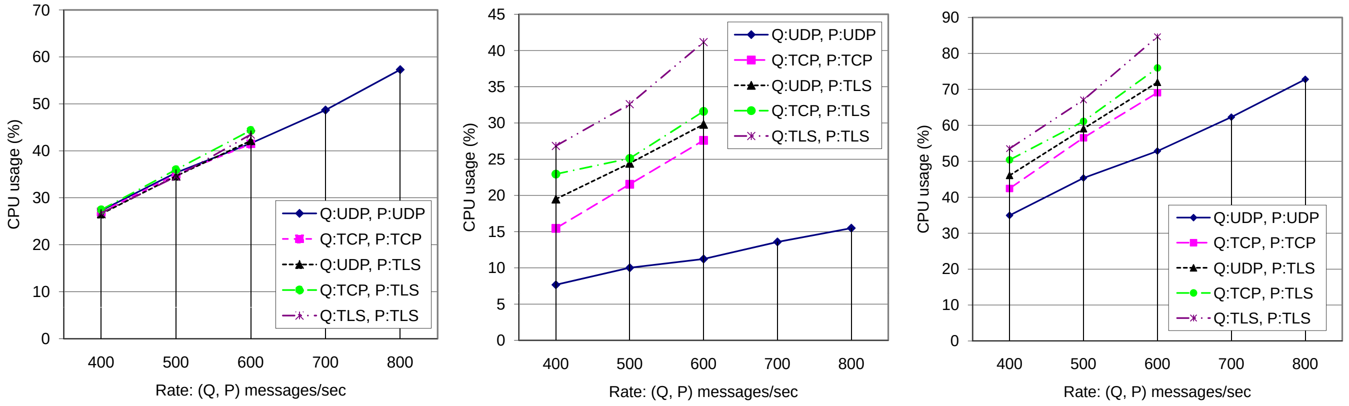
D. Delay

Since a sender has to get permission before sending data packets, there is one round-trip delay before sending the data packets.

The one-hop delay of Query and Permission message deliver, T_{query} and $T_{permission}$, are

$$\begin{aligned} T_{query} &= T_{gist} + T_{qtx} + T_{qpr} \\ T_{permission} &= T_{ptx} + T_{ppr} \end{aligned}$$

where T_{gist} is the GIST handshake delay, T_{qtx} is the sum of transmission delay and propagation delay of the Query message, T_{qpr} is the Query message processing delay , T_{ptx} is the sum of transmission and propagation delay of the Permission message, and T_{ppr} is the Permission message processing delay.



(a) CPU usage of PBS NSLP

(b) CPU usage of GIST

(c) CPU usage of NSIS

Fig. 8. CPU usage of PBS at a router when RSA-1024 is used for public key cryptography. The X-axis represents the rate of Query and Permission message pairs per second. For example, 400 means that the router handles 400 Query messages and 400 Permission messages per second.

There is no GIST handshake delay in the Permission message delivery since GIST handshake was already performed during the delivery of the Query message.

Thus, the total round-trip time, RTT, is

$$RTT = (T_{query} + T_{permission}) \times (N_r + 1)$$

where N_r is the number of routers that have a PBS functionality.

TABLE II

PROCESSING DELAY IN MSEC WHEN RSA-1024 IS USED FOR AUTHENTICATION FIELD —— THE FIRST ROW OF THE TABLE REPRESENTS THE TRANSPORT LAYER PROTOCOL TO DELIVER SIGNALING MESSAGES

	UDP	TCP	TLS
GIST handshake delay	0.411	10.057	23.383
Processing delay (Query)	0.423	0.429	0.523
Processing delay (Permission)	0.436	0.429	0.523

Since the propagation delay depends on the physical link and the transmission delay depends on the network link, we focus on the processing delay which depends on the PBS functionality, including cryptography algorithm. Since signaling messages have the authentication field generated by the public key, the verification of the authentication at a router increases the processing delay.

Table II shows GIST handshake delay and the PBS NSLP processing delay by protocols. GIST handshake delays are different by protocols since TCP requires a TCP session setup delay and TLS requires additional TLS session handshakes. Processing delays when UDP and TCP are used are almost same. However, when TLS is used, since the message is encrypted by a session key, decryption is required for signaling messages. Thus, the processing delay when TLS is used is the largest.

VI. Deployment and Application

To effectively prevent attacks, at least one PBS router should be placed between the attacker and the receiver. If the edge router in the receiver's area has PBS functionality, the edge router can properly handle the attacks even though the attack packets are generated in the backbone area. If the edge router in the attacker's area has PBS functionality, the attacks can be prevented near the attacker. Thus, installing the PBS functionality at the edge router can effectively prevent the attacks that are generated in the middle of the path. However, the current PBS system is unable to handle the case where the edge router in the receiver's area is compromised. If the compromised edge router starts sending attack packets, these would reach the receiver.

PBS is based on deny-by-default. Thus, in close-networks in which all end-users have PBS functionality, the packets that do not have permission will be dropped at the router. However, for short stream flows, such as DNS and ICMP, the cost in terms of flow state setup delay and signaling message overhead is high. Thus, PBS is not applicable to these short stream flows. The short stream flows can be rate-limited, and the rate-limiting can reduce the effect of attacks by flooding the short streaming data.

In the open-networks in which some end users do not have PBS functionality, the packets that do not have PBS functionality will be rate-limited. Therefore, those packets without permission will be treated as short stream flows.

VII. Related Work

Capability-based mechanisms, such as Capabilities [7], SIFF [8], TVA [9] and Portcullis [10], are similar to PBS since they use explicit permission. Data packets carry a capability, a collection of path specific information that is inserted by routers using a keyed hash algorithm. The capability is granted by the receiver. Thus, the packets that do not have the capability are dropped at routers. However, if an off-path attacker who spoofs a legitimate sender's address is in the same subnet and obtains the capabilities through eavesdropping, the attacker can inject the attack packets using the capabilities. Furthermore, these mechanisms can be compromised by the on-path attacker. Compromised routers can announce the capability to the upstream nodes, so the nodes can use the capability. Furthermore, the compromised router can use the capability to inject the attack flow and it can drop legitimate packets. Unlike PBS, there is no monitoring algorithm in capability-based mechanisms, these attacks cannot be detected and prevented. Since PBS uses on-path signaling for permission state setup at routers, PBS supports the easy installation and management of the permission state. In addition, the periodic state refresh through on-path signaling in PBS supports robustness of the system from state changes. However, capability-based mechanisms are weak during state changes.

Pushback [3] is an aggregate-based filtering mechanism. A router asks upstream routers to rate-limit an aggregate that causes congestion of the link. It cannot differentiate attack packets and poor packets (that are under the attack). If the legitimate packets and attack packets are in the same edge network, the legitimate packets are also dropped by rate-limiting.

StopIt [4] is filter-based DoS defense system. It installs filters at StopIt servers through request messages when there is an attack. If a StopIt server discovers that a sender is misbehaving, it sends a StopIt request to the sender, and a StopIt server will filter the flow from the sender. However, it requires modifying BGP to announce the StopIt servers' addresses. It does not effectively prevent on-path attacks, especially black hole attacks. Furthermore, the flooding attacks have already affected the links and the destination before the filtering is installed at StopIt servers.

NUTSS [25] presents an architecture for flow establishment using off-path and on-path signaling. NUTSS mainly considers naming, addressing, and middle box configuration. Even though NUTSS also presents a scheme to prevent attacks, it uses capabilities for this. Thus, it has the same problems as capability-based approaches.

VIII. Conclusion

We developed PBS, a signaling architecture for network traffic authorization. PBS is the hybrid of proactive (explicit permission) and reactive approach (monitoring network traffic for attacks) to prevent DoS attacks. PBS supports secure permission state setup and management, robustness against route changes. PBS works on the current networking architecture using existing transport protocols (UDP or TCP) and security protocols (public key cryptography, TLS/DTLS, IPsec).

We show that the PBS detection algorithm (PDA) can efficiently detect on-path and off-path attacks regardless of network type (trustworthy or Byzantine networks). Based on the detected and identified attacks, PBS suggests solutions, such as using stronger cryptography algorithm for IPsec or changing the data path, to the senders that are affected by the attack. Our analysis shows that PBS can prevent various kinds of attacks and the performance evaluation shows that PBS supports scalability.

IX. Acknowledgment

This work is supported by a grant from InterDigital Communications, LLC.

References

[1] Avaliable: http://www.symantec.com/, Symantec Corporation.

[2] Avaliable: http://www.arbornetworks.com/, Arbor Networks.

[3] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 62–73, 2002.

[4] X. Liu, X. Yang, and Y. Lu, "To Filter or to Authorize: Network-Layer DoS Defense Against Multimillion-node Botnets," in *Proceedings of ACM SIGCOMM*, Seattle, WA, USA, August 2008.

[5] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proceedings of ACM SIGCOMM*, Stockholm, Sweden, August 2000.

[6] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: secure overlay services," in *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[7] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet denial-of-service with capabilities," in *Proceedings of ACM Hotnets-II*, Cambridge, MA, USA, November 2003.

[8] A. Yaar, A. Perrig, and D. Song, "SIFF: a stateless Internet flow filter to mitigate DDoS flooding attacks," in *Proceedings of the IEEE Security and Privacy Symposium*, Oakland, CA, USA, May 2004.

[9] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting network architecture," in *Proceedings of ACM SIGCOMM*, Philadelphis, PA, USA, August 2005.

[10] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu, "Portcullis: protecting connection setup from denial-of-capability attacks," in *Proceedings of ACM SIGCOMM*, Kyoto, Japan, August 2007.

[11] Avaliable: http://www.ietf.org/html.charters/nsis-charter.html/, Next Steps in Signaling (nsis) working group.

[12] R. Hancock, G. Karagiannis, J. Loughney, and S. V. den Bosch, "Next Steps in Signaling (NSIS): Framework," Internet Engineering Task Force: RFC 4080, June 2005.

[13] H. Schulzrinne and R. Hancock, "GIST: General Internet Signaling Transport," Internet Engineering Task Force, Internet Draft, March 2009, work in progress.

[14] X. Fu, H. Schulzrinne, H. Tschofenig, C. Dickmann, and D. Hogrefe, "Overhead and Performance Study of the General Internet Signaling Transport (GIST) Protocol," in *Proceedings of IEEE Infocom*, Barcelona, Spain, April 2006.

[15] S. Hong and H. Schulzrinne, "PBS NSLP: Network Traffic Authorization," Internet Engineering Task Force, Internet Draft, October 2008, work in progress.

[16] R. Beverly and S. Bauer, "The spoofer project: inferring the extent of source address filtering on the Internet," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, Berkeley, CA, USA, 2005.

[17] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of Internet background radiation," in *Proceedings of the ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2004.

[18] A. Juels and J. Brainard, "Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, California, USA, February 1999.

[19] T. Aura, P. Nikander, and J. Leiwo, "Dos-resistant authentication with client puzzles," in *Lecture Notes in Computer Science*. Springer-Verlag, 2000, pp. 170–177.

[20] Avaliable: http://user.informatik.uni-goettingen.de/ nsis/, open Next Steps In Signaling (OpenNSIS) Implementation.

[21] Avaliable: http://www.openssl.org/, OpenSSL.

[22] Avaliable: http://www.netfilter.org/, Netfilter.

[23] O. Elkeelany, M. Matalgah, K. Sheikh, M. Thaker, G. Chaudhry, D. Medhi, and J. Qaddour, "Performance analysis of IPSec protocol: encryption and authentication," in *Preecidings of IEEE International Conference on Communications (ICC)*, New York, NY, April 2002.

[24] D. Lee and N. Brownlee, "Passive measurement of one-way and two-way flow lifetimes," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 17–28, 2007.

[25] S. Guha and P. Francis, "An end-middle-end approach to connection establishment," in *Proceedings of ACM SIGCOMM*, Kyoto, Japan, August 2007.