SECURITY SOLUTIONS

Network Address Translators:

Effects on Security Protocols and Applications in the TCP/IP Stack

Shiuh-Pyng Shieh, Fu-Shen Ho, Yu-Lun Huang, and Jia-Ning Luo
National Chiao Tung University, Taiwan

Some widely deployed protocols work transparently within NAT environments, but others fail completely or require special solutions to NAT-enable them. As NATs proliferate, it is important to recognize where they can be used without breaking the protocols used in the networks.

One proposed method for mitigating the address shortage problem in IPv4 is to use network address translators (NATs) to allow address reuse. The basic idea is to transparently map a wide set of private network addresses and corresponding TCP/UDP ports to a small set of globally unique public network addresses and ports.

NAT devices provide a way to handle IP address depletion incrementally—without changing hosts and routers—until more long-term approaches like IPv6 can be implemented. Existing Internet security protocols must be re-examined, however, to see how they function within this new network environment. We begin with a description of the four NAT environments and a discussion of their limitations. We then examine the relationships between NAT devices and popular Internet security protocols and applications at each layer of the TCP/IP stack to see if they can survive with NAT devices.

NAT ENVIRONMENTS

Figure 1 shows a NAT router with two interfaces. The device provides transparent routing between an intranet (using private IP addresses such as 10.1.1.1) and the Internet (using public IP addresses such as 140.113.215.1). Host addresses in the private network are unique only within the network, so the router converts unregistered internal addressing schemes to registered addresses before forwarding packets to public networks.

There are four common NAT environments defined in RFC 2663. With *traditional* NAT, hosts within private networks can unidirectionally access remote hosts in external networks. External network hosts, however, cannot initiate session requests to hosts inside private networks.

A *bidirectional* NAT server (also called two-way NAT), allows both inbound and outbound sessions. Once a connection is established in either direction, the NAT server maps the private network address statically or dynamically to a globally unique address. Bidirectional NAT assumes that fully qualified domain names for hosts in private and public networks are end-to-end unique. A DNS application-level gateway (ALG) must there-

1089-7801/00/$10.00 ©2000 IEEE

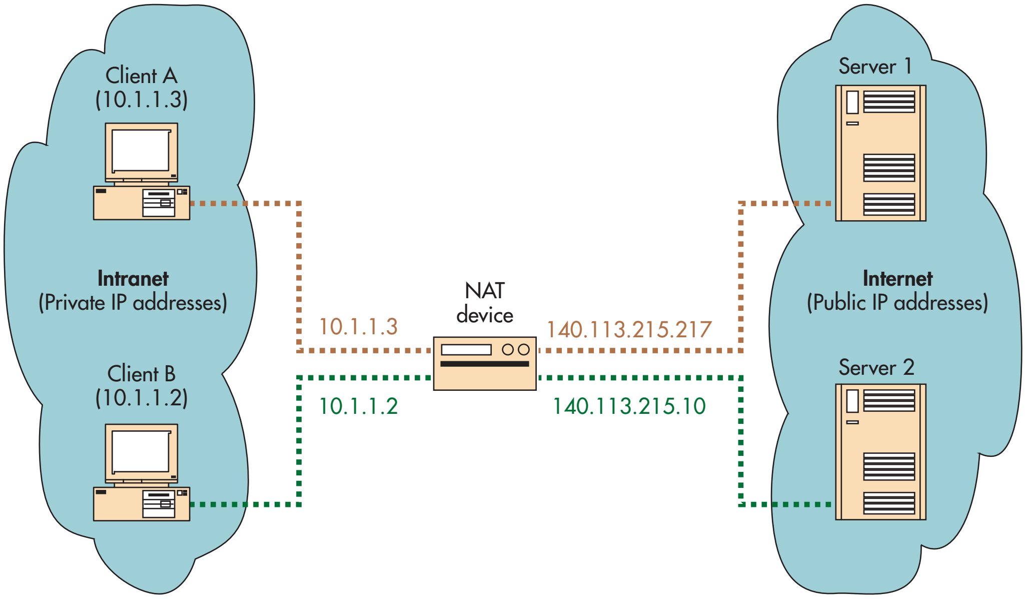

Figure 1. Example network using NAT. The NAT router maintains a mapping pool for address translation and routing purposes. Hosts with private addresses (10.x.x.x) are assigned temporary public addresses (140.x.x.x) when they connect to the Internet.

fore be used with bidirectional NAT to facilitate name-to-IP address and TCP/UDP port mappings.

Twice NAT modifies both the source and destination addresses for packets in a NAT session—unlike traditional and bidirectional NAT, which modify just one. Twice NAT is typically used when there are conflicts in the address space of both source and destination network addresses. In addition to the translation of source addresses for outbound packets, the twice-NAT server maps the external network host's registered IP address to another unique private IP address.

Finally, *network address and port translation* (NAPT) extends NAT a step further by also translating source and destination port numbers and checksum values in TCP/UDP protocol headers in the transport layer. NAPT also lets NAT servers modify these transport identifiers in a way that is generally transparent to upper-layer protocols and does not affect their behaviors. Address translation becomes difficult, however, when a payload containing IP address and port information is encrypted by security protocols because the server cannot decrypt the payload.

LIMITATIONS AND SECURITY

Despite the convenience that address translation brings, it also has limitations. Not all applications can pass through NAT servers transparently, and those carrying IP address and TCP/UDP port information inside their payloads require ALGs for both outbound and inbound sessions. By breaking the end-to-end nature of Internet applications, NAT threatens Internet security. Some existing security protocols fail, for example, when address translation is applied to authentication packets. There are, however, a few methods for minimizing this drawback for many such protocols.

In Figure 2, for example, all payloads routed by the NAT server are forwarded to the ALG, which interprets them and performs the necessary address translations on the connection. If a session is initiated from the public network, however, the ALG must be integrated with the NAT server in order to allow inbound sessions to pass through the server.

The server can also perform translation with port forwarding for some simple protocols that use only fixed ports.[1] This feature lets us forward all packets from certain ports to their dedicated servers, while treating the NAT as a virtual server that distributes traffic among designated server farms. This solution still cannot provide end-to-end security, however, especially when the NAT server and the ALG are outside a trusted boundary. An ALG can also become a bottleneck that considerably degrades forwarding throughput for the border router and NAT server.

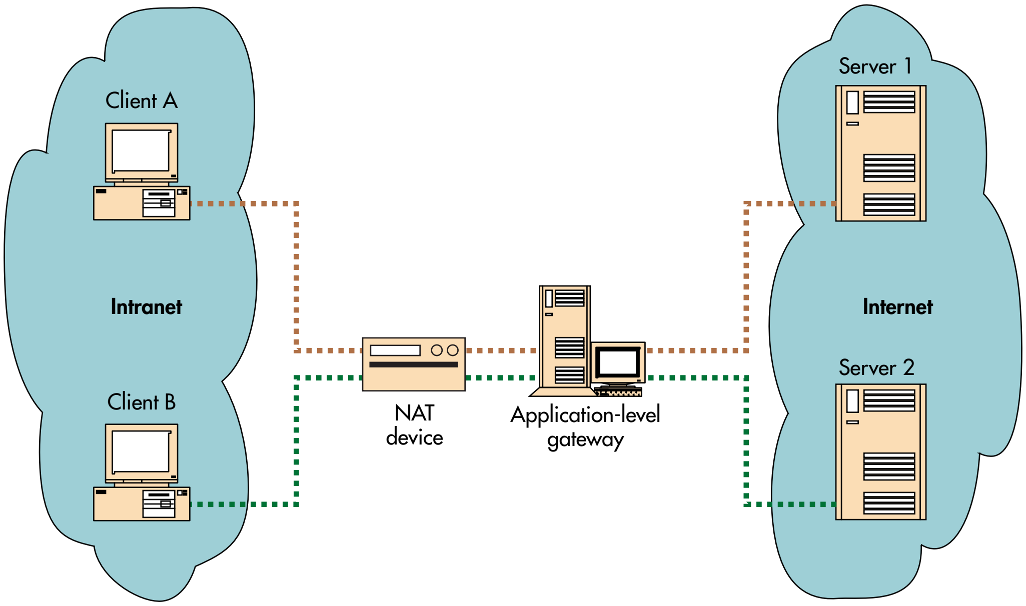

Figure 2. Example network with NAT and ALG. An application-level gateway intercepts protocol payloads and performs necessary address translation between the private and public networks. ALGs are usually application-dependent.

Furthermore, NAT servers must maintain state information for modified addresses while communicating with external hosts in order to ensure the datagrams in the session are routed to the correct destination at either end. A NAT router maintains mapping relations for all sessions established through it, and requests and responses for those sessions must be routed back through it as well. The NAT router should thus be combined with a border router in a domain for better performance.

Regardless of the method used, NAT must be able to translate headers and packets according to the referred addressing scheme. Many security protocols exchange IP addresses or TCP/UDP port-related information in their authentication packets, which makes them vulnerable when passing through NAT servers. Any host applying encryption to TCP/IP checksums must thus be assigned a globally unique IP address that is exempted from address translation.

NETWORK AND TRANSPORT LAYER PROTOCOLS

Address translation is meaningful only at the network layer and above. NAT servers in the network layer translate all IP addresses from private to public, and the original checksum in the packet header becomes incorrect once the source or destination address is modified. The NAT server's basic function, aside from IP translation, is to recalculate the checksum after the modification. Using a NAT device can cause problems when encryption is applied to the IP address or port number-related protocol data at these two layers using a signed or modification-proof function. In the following, we discuss how NAT affects some important network and transport layer protocols.

Virtual Private Networks

Virtual private network technology has become a popular solution for enterprises looking to secure their intranets, and three major tunneling protocols have been proposed for building a VPN: the point-to-point tunneling protocol (PPTP, RFC 2637), layer-2 tunneling protocol (L2TP, RFC 2661), and the IP security protocols (IPSec, RFCs 2401, 2402, 2406).[1]

Tunneling protocols. As long as addresses are globally unique, PPTP and L2TP can both transparently provide users with end-to-end services when tunneling between border routers. Neither protocol can provide end-to-end tunnels, however, when one of the endpoints is behind a NAT server. Enterprises will encounter problems when combining VPN technology with NAT devices to establish tunnels between an endpoint in a private domain and another in a private or public domain.

IPSec. IPSec defines two packet headers: the *authentication header* (AH) for handling authentication, and the *encapsulating security payload* (ESP) for encryption. The AH helps ensure data integrity by preventing packet modification by a third party. Unfortunately, it also prohibits NAT servers from performing address translation. Not even an ALG can modify the authentication header because it does not know the source and destination hosts' secret information to reproduce the authentication header after the modification. Moreover, TCP/UDP port numbers and checksums are embedded in the ESP, and may sometimes be encrypted. A NAT server can use a traditional translation method for packets with ESP headers, but a server using NAPT cannot modify ESP-enabled packets without the secret key to decrypt and rebuild the ESP. In addition to these problems, IPSec's third main component, key management, is also incompatible with NAT.

There are two ways to handle key exchange and key management in IPSec: *manual keying*, which is suitable with a small number of hosts, and *automated keying*. A scalable automatic key management protocol, such as the Internet key exchange (IKE), is required for on-demand creation of a security association (SA).

The SA is used to bundle IPSec with the cryptographic algorithms and key exchange method agreed on by the source and destination endpoints. An SA must be derived using key exchange before an IPSec connection can be established, and the packets transmitted during the process contain encrypted IP addresses and authentication values, which the NAT server cannot modify. Thus, IKE cannot pass through NAT devices.

As an alternative to IKE, some enterprises use another distribution scheme, simple key management for IP (SKIP, RFC 2409), for encryption key exchange in VPNs.[2] Rather than session-oriented keys, SKIP uses packet-oriented keys, which are communicated in-line. The NAT server can translate the standard IP header with traditional methods. When NAPT is used, however, the NAT server cannot translate the payload because it is encrypted using the packet key and may contain TCP/UDP port numbers.

Socket Layer Protocols

As shown in Figure 3, the socket layer is a logical layer between the transport and application layers in the TCP/IP stack. It encapsulates the transport layer for programming and manages a logical connection between two endpoints to simplify access to underlying layers. Only a few Internet security protocols, such as secure socket layer (SSL)[3] and transport layer security (TLS, RFC 2246), are currently deployed in this layer.

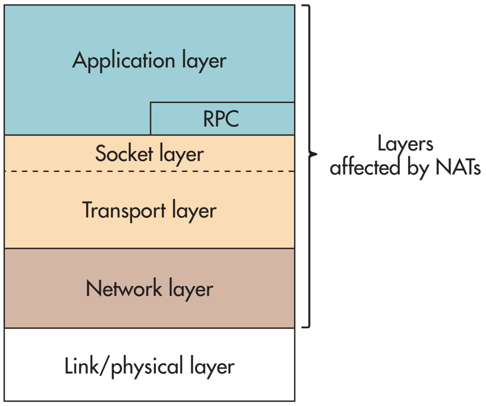

Figure 3. TCP/IP layer stack. NAT is meaningful only at and above the physical network layer.

The SSL protocol is designed to provide communication privacy over the Internet by preventing message forgery and eavesdropping. SSL intercepts messages transmitted from the application layer and fragments them into blocks. The protocol can then compress the data before applying the Message Authentication Code, encrypting the messages, and transmitting the result to the transport layer. Because SSL does not use IP or port information to verify a user's identity, it is network-independent and works equally well with both traditional NAT and NAPT.

As a successor of SSL, TLS is also designed to provide end-to-end security over the Internet. TLS also verifies user identification without depending on IP and port information, which lets it pass transparently through NAT and NAPT.

APPLICATION LAYER PROTOCOLS

In this section, we examine how several primary groups of application layer protocols interact with NAT. Encryption, signed, or modification-proof functions applied to IP address or port-number-related data cause similar problems with NATs at the application layer to those at the network and transport layers. Application-layer protocols also commonly use dynamic port number assignment in protocol sessions, which can cause problems for NATs. In the following sections, we will discuss some popular application layer protocols and their issues with NATs.

Remote Procedure Calls

The remote procedure call (RPC, RFC 1831) is treated as middleware by application layer protocols between the transport and application layers with the aid of RPC protocol tools, such as *rpcgen* in Unix-like systems. Many applications, such as the network file system (NFS), rely on the RPC model for distributed interaction because it uses an application-oriented approach and relates client-server communication to conventional procedure calls.

RPC normally uses three different UDP ports to set up a session, and the packets exchanged between an RPC client and server during the setup process contain IP address and port information. As a result, protocols that utilize RPC as an underlying layer fail when used with NAT. Not even an ALG can resolve the problem because some RPC packets contain encrypted information and cannot be modified by the NAT server.

Authentication Protocols

NAT devices are usually deployed in enterprise networks, for which authentication plays an important role in Internet security. Because most Internet authentication protocols authenticate users or verify certificates by users' privately held information only, these protocols can usually pass through NAT servers with the aid of DNS-ALG or port forwarding.

Kerberos. The Kerberos authentication protocol, designed for TCP/IP networks by B.C. Neuman et al. at MIT, uses a combination of packet encryption, time-based credentials, and a trusted third party to provide secure authentication.[4] If a client in a private realm initiates requests to a ticket-granting server, authentication server, or destination server located in a public network, address translation can be performed with no problem—unless any one of these servers is located in a different private network than the client's realm. In this case, DNS-ALG or port-forwarding technology is required for the authentication process to succeed because the client can't directly reach servers hidden by the NAT server.

Radius. The remote authentication dial-in user service protocol (Radius, RFC 2138) carries authentication, authorization, and configuration information between a network-access server (NAS) and a shared authentication server. The NAS operates as a Radius client and passes user information to the server. The connection between the client and Radius server is authenticated by a shared secret rather than by IP address, so authentication is unaffected by the NAT server as long as the Radius server is located on the public network. If the Radius server is seated on a private network and unreachable by outside clients, DNS ALG or port forwarding will still allow authentication packets to successfully pass through the NAT server.

S/Key. S/Key (RFC 2289) limits the use of any password to a single communication session.[5] When a user logs in, the S/Key-enabled network server issues a challenge consisting of a number and a string of characters for the user to calculate the one-time password. S/Key packets contain no IP or port information, so as with Radius, authentication fails to traverse the NAT server if the S/Key server is located on a private network unless DNS ALG or port forwarding is used.

Voice over IP

International Telecommunication Union recommendation H.323 defines the components, procedures, and protocols for packet-based audiovisual communication, and it is fundamental to the growth of voice over IP.[6] H.323 uses multiple control sessions to negotiate the IP addresses and port numbers for successive H.235 authentication and Real-time Transport Protocol (RFC 1889) audio or video sessions.

When passing through a NAT server, the successive sessions fail because the server has no knowledge about the encoded payloads, which include the addresses and port numbers. In this case, an application-specific ALG (called an H.323 proxy) is necessary between the calling and called parties to look into the H.323 messages and perform the address and port translation, so that the NAT server can be prepared for the successive sessions.

The session initiation protocol (SIP, RFC 2543) and media gateway control protocol (MGCP, RFC 2705) are other key application-layer protocols for VoIP. Both use the session description protocol (SDP, RFC 2327) to derive the RTP address and port number for voice communication. Like H.323, both SIP and MGCP require an ALG between the parties in order to perform the translation.

Secure Electronic Commerce

Many protocols have been proposed for transactions and payments in Internet-based commerce, but the secure electronic transaction (SET) protocol is currently the most popular solution.[7] With SET, cardholders and merchants must obtain their own certificates from a trusted certificate authority prior to transacting business. User identification, credit card information, and electronic purchase orders can be transmitted securely over the Internet using the public key cryptosystem and the certificates for the parties involved in the transaction. Because the SET payload contains no IP addresses or port information, it works under all NAT environments.

Other Popular Protocols

Address translation is directly affected by the characteristics of some of the most popular Internet protocols, including the file transfer protocol (FTP), mailer protocols, and secure shell (SSH). No discussion of the relationship between NAT devices and protocols would be complete without examining these.

FTP. FTP does not work under traditional NAT or NAPT, although its authentication commands work fine in NAT environments. The PORT and PASV commands both include IP address and port numbers to enable the destination host to set up another data connection, and the NAT server cannot read the encapsulated information on its own. An FTP ALG (also called an FTP proxy) must be used to analyze the payloads in order for FTP to work with traditional NAT, NAPT, or bidirectional NAT servers.

FTP data is often fragmented into packets of various lengths, which must be reassembled before address translation can be performed. To modify the fragmented packets, the FTP ALG maintains TCP/UDP state information and regenerates the original packets. If the replaced packet is longer than the original, the ALG must split the replacement into fragments again and modify all the packet sequence numbers.

Mail protocols. Most popular mail protocols, such as SMTP (RFC 876) and POP3 (RFC 1939), can pass through NAT servers as long as port forwarding is applied because they do not include IP addresses in the data payload. Version four of the Internet message access protocol (IMAP4, RFC 2060), however, uses mechanisms such as Kerberos and S/Key to authenticate users accessing e-mail or bulletin board messages from mail servers. As noted, some of these mechanisms are not NAT-compatible, unless a DNS ALG or port forwarding is used when the IMAP4 server is located behind a NAT device.

Secure shell. SSH was designed to replace well-known Internet communication applications like telnet and rlogin for remote login and other secure network services over the Internet. To prevent transmitting clear data across the insecure network, SSH uses public-key-based cryptosystems for authenticating users and verifying certificates. Authentication is based on user identity, rather than IP address and port information, so like other certificate-based protocols, SSH works well in NAT environments.

COMPARISONS

Table 1 (page 48) summarizes the NAT-compatibility of the protocols described in this article. The comparison assumes that clients are located on private networks and servers are on public networks, which is the most common topology used with NATs.

In many cases, placing ALGs outside the private network can NAT-enable protocols that use embedded IP addresses and port numbers. Protocols that encrypt packets with IP addresses and port numbers can be handled by exchanging decryption keys between authentication servers and NAT servers, but this solution is not generally practical because NAT servers are usually located at network boundaries, where they can become the target of attacks. Once the NAT server is compromised, an intruder can obtain all secret information passed through it. In addition, all traffic goes through the NAT server, and performing encryption functions there can heavily degrade throughput. To date, no good solution exists for this problem.

Another serious problem occurs when the protocols listed in Table 1 are run on servers located on private networks: None of the clients can reach them. A general solution to this problem, which applies only to fixed-port protocols, is to add port-forwarding or ALG functions to the NAT servers. Some modern routers supporting NAT also include built-in ALGs and port-forwarding functions for working with popular protocols.

Because they use only a limited number of global IP addresses, architectures in which client and server are both located in private realms are likely to increase in popularity as cable-modem or xDSL-based enterprise networks are deployed. With this type of architecture, we can try to use the solutions described above and derive hybrid solutions. For example, combining an ALG and port-forwarding functions can create a solution for a given protocol. Without NAT standards, however, the solutions remain application-dependent.

FUTURE WORK

Address space will no longer be an issue once a solu-

Table 1. Compatibility of common Internet protocols with NAT environments.

	NAT Environments					
Protocols	NAT	Two-way NAT	Twice NAT	NAPT	Reason for failure	ALG as solution
Network and Transport Layer						
PPTP	Yes	Yes	Yes	Yes	N/A	N/A
L2TP	Yes	Yes	Yes	Yes	N/A	N/A
IKE	No	No	No	No	Encrypted IP address	No
SKIP	Yes	Yes	Yes	No	Encrypted port number	No
SSL	Yes	Yes	Yes	Yes	N/A	N/A
TLS	Yes	Yes	Yes	Yes	N/A	N/A
Application Layer						
RPC	No	No	No	No	1. Dynamic port numbers 2. Encrypted IP address and port numbers	1. Yes 2. No
Kerberos	Yes	Yes	Yes	Yes	N/A	N/A
Radius	Yes	Yes	Yes	Yes	N/A	N/A
S/Key	Yes	Yes	Yes	Yes	N/A	N/A
H.323	No	No	No	No	Dynamic IP address and port numbers	Yes
SIP	No	No	No	No	Dynamic IP address and port numbers	Yes
MGCP	No	No	No	No	Dynamic IP address and port numbers	Yes
SET	Yes	Yes	Yes	Yes	N/A	N/A
FTP	No	No	No	No	IP address and port number in FTP commands	Yes (FTP-ALG)
SMTP	Yes	Yes	Yes	Yes	N/A	N/A
POP3	Yes	Yes	Yes	Yes	N/A	N/A
IMAP4	Yes	Yes	Yes	Yes	N/A	N/A
SSH	Yes	Yes	Yes	Yes	N/A	N/A

tion like IPv6 gets widely deployed on the Internet, but NAT-based solutions appear to be here for the near future. Indeed, Internet service providers might provide only NAT solutions to small enterprise networks due to the shortage of addresses in IPv4. Given this trend, existing Internet security protocols must be re-examined to see how they function within NAT network environments.

Some popular Internet security protocols can survive when deployed in NAT environments, but others fail completely or require complicated solutions. When IPSec is used in a network with NAT, for example, there is no solution that does not break end-to-end security—not even an ALG—because the authentication headers and ESPs prohibit modifications to the IP headers and payloads. Other protocols require modifications and ad hoc work-arounds to be NAT-friendly. For instance, authentication protocols can work with NAT devices as long as IP addresses and port numbers are not sent in encrypted, signed, or hashed messages. Identifiers that are unrelated to such information should be used for routing instead.

ALGs can also be used for address translation with some protocols, but they raise security and bottleneck issues that deserve further studies. Other solutions can sometimes be found for specific NAT implementations, but the lack of common translation rules can keep these solutions from working in other NAT implementations. Efforts toward establishing common, negotiable NAT rules for all implementations to follow could help ensure that higher-layer protocols can survive under NAT environments. ■

ACKNOWLEDGMENTS

This work is supported in part by the Ministry of Education and National Science Council of Taiwan. The authors especially thank and acknowledge the anonymous referees for their suggestions and comments on this article.

REFERENCES

1. D. Kosiur, *Building and Managing Virtual Private Networks*, John Wiley & Sons, New York, 1998.
2. A. Aziz, M. Patterson, and G. Baehr, "Simple Key-Management for Internet Protocols (SKIP)," *Proc. 1995 Int'l Networking Conf.,* Internet Soc., Reston, VA; available at http://www.isoc.org/HMP/PAPER/244/abst.html.
3. A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol," specification, Netscape Comm. Corp., Nov. 1996.
4. B.C. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Comm.*, vol. 32, no. 9, Sep. 1994, pp. 33-38; available at http://www.isi.edu/gost/publications/kerberos-neuman-tso.html.
5. N. Haller, "The S/KEY One-Time Password System," *Proc. Internet Soc. Symp. Network and Distributed System Security*, Internet Soc., Reston, VA, 1994, pp. 151-158.
6. "Series H: Audiovisual and Multimedia Systems, Infrastructure of Audiovisual Services—Systems and Terminal Equipment for Audiovisual Services," ITU-T Recommendation H.323, Feb. 1998.
7. "Secure Electronic Transaction (SET) Specification—Book 3: Formal Protocol Description, version 1.0," Visa and MasterCard, May 1997; available online at http://www-s2.visa.com/nt/ecomm/set/set_bk3.pdf.

Shiuh-Pyng Shieh is director of the Computer and Network Center and a professor with the Department of Computer Science and Information Engineering at National Chiao Tung University, Taiwan. He has an MS and a PhD in electrical engineering from the University of Maryland, College Park. His research interests include internetworking, distributed operating systems, and network security.

Fu-Shen Ho received a BS and an MS in computer science and information engineering from National Chiao Tung University. He is currently a PhD candidate in the same department. His research interests include operating systems design, distributed systems, network security, and VoIP systems.

Yu-Lun Huang received a BS in computer science and information engineering from National Chiao Tung University. She is currently a PhD candidate in the same department. Her research interests include electronic commerce, distributed systems, data hiding, network security, and VoIP systems. She is a member of the Phi Tau Phi Society.

Jia-Ning Luo received a BS in electrical engineering and an MS in computer science from Ta-Tung University. He is currently a PhD candidate at National Chiao Tung University. His research interests include computer networks, Internet security, and e-commerce.

Readers can contact the authors at (ssp, fsho, ylhuang, jnluo)@csie.nctu.edu.tw.

IETF Requests For Comments

The RFCs referenced in this article are available online at the IETF RFC editor at http://www.ietf.org/rfc/rfcxxxx.txt.

RFC 876 • "Survey of SMTP Implementation," D. Smallberg, Sept. 1983.

RFC 1831 • "RPC: Remote Procedure Call Protocol Specification, version 2," R. Srinivasan, Aug. 1995.

RFC 1889 • "RTP: A Transport Protocol for Real-Time Applications," H. Schulzrinne et al., Jan. 1996.

RFC 1939 • "Post Office Protocol, version 3," J. Myers and M. Rose, May 1996.

RFC 2060 • "Internet Message Access Protocol, version 4 revision 1," M. Crispin, Dec. 1996.

RFC 2138 • "Remote Authentication Dial-In User Service (RADIUS)," C. Rigney et al., Apr. 1997.

RFC 2246 • "The TLS Protocol, version 1.0," T. Dierks and C. Allen, Jan. 1999.

RFC 2289 • "A One-Time Password System," N. Haller et al., Feb. 1998.

RFC 2327 • "SDP: Session Description Protocol," M. Handley and V. Jacobson, Apr 1998.

RFC 2341 • "Cisco Layer-Two Forwarding (Protocol) (L2F)," A. Valencia, M. Littlewood, and T. Kolar, May 1998.

RFC 2401 • "Security Architecture for the Internet Protocol," S. Kent and R. Atkinson, Nov. 1998.

RFC 2402 • "IP Authentication Header," S. Kent and R. Atkinson, Nov. 1998.

RFC 2406 • "IP Encapsulating Security Payload (ESP)," S. Kent and R. Atkinson, Nov. 1998.

RFC 2409 • "The Internet Key Exchange (IKE)," D. Harkins and D. Carrel, Nov. 1998.

RFC 2543 • "SIP: Session Initiation Protocol," M. Handley et al., Mar. 1999.

RFC 2637 • "Point-to-Point Tunneling Protocol (PPTP)," K. Hamzeh et al., July 1999.

RFC 2661 • "Layer-Two Tunneling Protocol (L2TP)," W. Townsley et al., Aug. 1999.

RFC 2663 • "IP Network Address Translator (NAT) Terminology and Considerations," P. Srisuresh and M. Holdrege, Aug. 1999.

RFC 2705 • "Media Gateway Control Protocol (MGCP) Version 1.0," M. Arango et al., Oct. 1999.