INTERNET SECURITY

BÜLENT YENER*
Rensselaer Polytechnic University
Troy, New York

1. INTRODUCTION

As the Internet utilized as a new commercial infrastructure, meeting security requirements of diverse applications becomes imminent. Furthermore, the Web and browsers bring have brought the Internet to homes of average people, creating not only a surge in use of the Internet but also a risk to their privacy.

Internet security aims to ensure *confidentiality, authentication, integrity*, and *nonreputiation* of the "information" carried over a collection of interconnected, heterogeneous networks via messages. Confidentiality or privacy prevents unauthorized parties from accessing the message. Authentication requires that source of a message has correct and verifiable identity. Integrity protection of information ensures that unauthorized parties cannot modify the information. Nonreputiation of information requires that the sender and receiver of the information cannot deny the transmission of the message. In general, the security attacks can be grouped into several classes:

1. *Interception* attacks, which are directed at the confidentiality of information by unauthorized access. It is a passive attack where the adversary simply observes the communication channel without modifying the information. Eavesdropping on a communication channel is a typical example.
2. *Modification* attacks, which violate the integrity of information. It is an active attack in which adversary changes the content of information. Man-in-the-middle attacks are typical examples.
3. *Fabrication* attacks, in which the adversary generates and inserts malicious information to the system. This is also an active attack and it violates the authenticity of information.
4. *Interruption*, which is also an active attack that targets the availability of the system. An example is malicious jamming in a wireless network to generate intentional interference.

1.1. Cryptography

Cryptography provides the essential techniques and algorithms to keep information secure [1,2]. Confidentiality is done by encryption and decryption. Authentication is ensured with digital certificates while integrity is protected with hash functions. Nonreputiation of messages is ensured with digital signatures.

The roots of cryptographic research can be traced to William F. Friedman's report *Index of Coincidence and Its Applications* [3], and Edward H. Hebern's rotor machine [4] in 1918. In 1948 Claude Shannon presented his work on the communication theory of secrecy systems in the *Bell System Technical Journal* [5]. In early 1970s work by Horst Feistel from IBM Watson Laboratory led the first U.S. Data Encryption Standard (DES) [6]. In DES both parties must share the same secret key of 56 bits before communication begins. However, Michael Weiner showed that exhaustive search can be used to find any DES key [7]. More recently, the National Institute of Standards and Technology (NIST) has selected the Advanced Encryption Standard (AES), the successor to the venerable DES. AES was invented by Joan Daemen and Vincent Rijmen [8].

1.1.1. Confidentiality. Encryption is a function E that takes plaintext message M as input and produces the encrypted ciphertext C of M: $E(M) = C$. Decryption is the function for the reverse process: $D(C) = M$. Note that $D(E(M)) = M$. In modern cryptography, a *key* K is used for encryption and decryption so that $D_K(E_K(M)) = M$. Cryptographic algorithms, based on using a single key, (i.e., both encryption and decryption are done by the same key) are called *symmetric ciphers* and they have two drawbacks: (1) an arrangement must be made to ensure that two parties have the same key prior to communicate with each other and (2) the number of keys required for a complete communication mash for an n party network is $O(n^2)$. Although a trusted third party such as a *key distribution center* (KDC) can be used to circumvent these two problems, it requires that KDC must be available in real-time to initiate a communication.

Public key cryptography proposed by Whitfield Diffie and Martin Hellman in 1975 [9] is based on *asymmetric ciphers*. In such systems, the encryption key K_1 is different from the decryption key K_2 so that $D_{K_2}(E_{K_1}(M)) = M$. The encryption key K_1 is called the *public key*, and it is not secret. The second key K_2 is called the *private key*, and it is kept confidential. The best known public key crypto system, proposed by Ronald Rivest, Adi Shamir, and Leonard Adleman (RSA) [10]. Although the public key ciphers reduce the number of keys to $O(n)$, they also suffer two problems: (1) key size is much larger than in symmetric systems and (2) the encryption and decryption are much slower. These two issues become problematic in a bandwidth processing-constrained environments such as wireless networks. Thus, the main use of public key systems is limited to distribution of symmetric cipher keys.

1.1.2. Integrity. Cryptographic solutions for integrity protection are based on *one-way hash functions*. A hash

* Department of Computer Science at Rensselaer Polytechnic Institute. This article was written in part while the author was visiting Bogazici University Department of Computer Science, Istanbul, Turkey.

Wiley Encyclopedia of Telecommunications, Edited by John G. Proakis
ISBN 0-471-36972-1

function is a one-way function that is easy to compute but significantly harder to compute in reverse (e.g., $a^x \bmod n$). It takes a variable-length input and produces a fixed-length hash value (also known as "message digest"). In general it works as follows. The sender computes the hash value of the message, encrypts it using the receiver's public key, and appends it to the message. The receiver decrypts the message digest using his/her private key and then computes the hash value of the message. If the computed hash value is the same as the decrypted one, the message integrity is considered to be preserved during transmission. However, this is not an absolute guarantee since a hash collision is possible (i.e., a modified or fabricated message may have the same hash value as the original. Furthermore, the hash function is public so that the attacker can intercept a message, modify it, and compute a new hash value for the modified message. Thus, it would be a good idea to encrypt the message as well or use a message authentication code (MAC), which is a one-way function with a key.

1.1.3. Nonreputiation. In order to prove the source of a message, one-way functions called *digital signatures* can be used in conjunction with public key cryptography. To stamp a message with its digital signature, the sender encrypts the message digest with its private key. The receiver first decrypts the message using the sender's public key and then computes the message digest to compare it to the one that arrives with the message.

1.1.4. Authentication. To prevent an attacker from impersonating a legitimate party, *digital certificates* are used. At the beginning of a secure Internet session, sender transmits its digital certificate to have his/her identity to be verified. Digital certificates may be issued in a hierarchical way for distributed administration. A digital certificate may follow the ITU standard X.509 [11,12] and is issued by a *certificate authority* (CA) as a part of public key infrastructure (PKI). The certificate contains the sender's public key, the certificate serial number and validity period, and the sender's and the CA's domain names. CA must ensure integrity of the issued certificate; thus it may encrypt the hash value of it using its private key and append it to the certificate.

1.2. Cryptography and Security

Although cryptography provides the building blocks, there is much to consider for Internet security. First, the rapid growth of the Internet increases its heterogeneity and complexity. A communication channel between a pair of users may pass through different network elements running diverse protocols. Most of these protocols are designed with performance considerations and carry design and implementation holes from security point of view. For example, consider the *Anonymous File Transfer Protocol* (FTP) [13], which provides one of the most important services in the Internet for distribution of information. There are several problems with FTP and its variants. For example, the *ftpd* deamon runs with superuser privileges for password and login processing. Thus, leaving a sensitive file such as the password file in the anonymous FTP site will be a serious security gap. Another example is the *Transport Control Protocol* (TCP) [14], which provides a connection between a pair of users. In TCP each connection is identified with a 4-tuple: <*source* (*local*) *host IP address, local port number, destination* (*remote*) *host IP address ID, remote port number*>. Since the same 4-tuple can be reused, the *sequence numbers* are used to detect the lingering packets from the previous uses of the tuple. There is a potential threat here since an attacker can "guess" the initial sequence number and convince the remote host that it is communicating with a trusted host. This is known as a *sequence number attack* [15]. A remedy for this attack would be to hide the target host behind a dedicated gateway called a *firewall* to prevent direct connections [16].

Also, the network software is hierarchical and layered. At each layer a different protocol is in charge and interacts with the protocols in the adjacent layers. In the following sections we will examine layer-specific security concerns.

2. LINK-LAYER SECURITY

Link-layer security issues in *local-area networks* (LANs) and *wide-area networks* (WANs) are fundamentally different. In a WAN link-layer security requires that end point of each link is secure and equipped with encryption devices. Although institutions such as the military have been using link-layer encryption, it is not feasible in the Internet.

In a LAN hosts share the same communication medium that has (in general) a broadcast nature (i.e., transmission from one node can be received by all others on the same LAN). Thus eavesdropping is easy and, to ensure confidentiality encryption, is required. For example, in a LAN it is better to use the SSH protocol [17] instead of Telnet to avoid compromising passwords.

2.1. Access Control

Typically, a LAN connects hosts who are in the same security or administrative domain. While allowing legitimate user accessing to a LAN from outside (via a dialup modem, or a DSL, or a cable modem), it is crucial to prevent unauthorized access. Firewalls are dedicated gateways used for access control and can be grouped into three classes [16]: packet filter gateways, circuit-level gateways, and application-layer gateways. Packet filters operate by selectively dropping packets based on source address, destination address, or port number. In a firewall the security policies can be specified in a table that contains the filtering rules to which the incoming or outgoing packets are subject. For example, all outgoing mail traffic can be permitted to pass through a firewall while Telnet requests from a list of hosts can be dropped. Filtering rules must be managed carefully to prevent loopholes in a firewall. In case of multiple firewalls managed by the same security domain, it is crucial to eliminate inconsistencies between the rules.

Application- and circuit-level gateways are firewalls that can secure the usage of a particular application

by screening the commands. Logically it resides in the middle of a protocol exchange and ensures that only valid commands are sent. For example, it may monitor a FTP session to ensure that only a specific file is accessed with read-only permission.

3. NETWORK-LAYER SECURITY

The Internet is composed of many independent management domains called *autonomous systems* (ASes). Internet routing algorithms within an AS and among ASes are different. Most important intra-AS routing (interior routing) and inter-AS (exterior routing) protocols are the Open Shortest Paths First (OSPF), and Border Gateway Protocol (BGP), respectively. However, the common theme in these protocols is the exchange of routing information to converge in a stable routing state. However, because of the lack of scalable authenticity check, routing information exchanged between the peers is subject to attacks. The attacker can eavesdrop, modify, and reinject the exchanged messages. Most of these attacks can be addressed by deployment of public key infrastructures (PKI) and certificates for authentication and validation of messages. For example, Kent et al. [18] discuss how to secure BGP protocol using PKI with X.509 certificates, and IPsec protocol suite. The solution proposes to use a new BGP path attribute to ensure the authenticity and integrity of BGP messages and validate the source of UPDATE messages. However, if a legitimate router is compromised, then such cryptographic mechanisms cannot be sufficient and the security problem degenerates to the Byzantine agreement problem [19] in distributed computing.

Next we discuss the IPsec protocols for securing IP-based intranets and then review the security issues in ATM networks.

3.1. IP Security:IPsec

The suite of IPsec protocols are designed to provide security for Internet Protocol version 4 (IPv4) and version 6 (IPv6) [20]. IPsec offers access control, connectionless integrity, source authentication, and confidentiality. IPsec defines two headers that are placed after the IP header and before the header of layer 4 protocols (i.e., TCP or UDP). These headers are used in two traffic security protocols: the *authentication header* (AH) and the *encapsulating security payload* (ESP). AH is recommended when confidentiality is not required, while ESP provides optional encryption. They both ensure integrity and authentication using tools such as keyed hash functions. Both AH and ESP use a simplex connection called a *security association* (SA). An SA is uniquely identified by a triple that contains a security parameter index (SPI), a destination IP address, and an identifier for the security protocol (i.e., AH or ESP). The negotiation of security association between two entities and exchange of keys can be done by using the Internet Key Exchange (IKE) protocol [22]. Conceptually, an SA is a virtual tunnel based on encapsulation. Two types of SAs are defined in the standard: *transport mode*, and *tunnel mode*. The former is a security association between two hosts, while the latter is established between network elements. Thus, in the transport mode the security protocol header comes right after the IP header and encapsulates any higher-level protocols. In the tunnel mode there is an outer header and an inner IP header. The *outer* header specifies the next hop, while the *inner* header indicates the final destination. The security protocol header in the tunnel mode is inserted after the outer IP header and before the inner one, thus protecting the inner header.

There are successful attacks on IPsec in spite of the secure ciphers used by the protocol. For example, consider the cut-and-paste attack by Bellovin [21]. In this attack, an encrypted ciphertext from a packet carrying sensitive (targeted) information is cut and pasted into the ciphertext of another packet. The objective is to trick the receiver to decrypt the modified ciphertext and reveal the information.

3.2. ATM Security

Asynchronous transfer mode (ATM) technology is based on establishing switched or permanent virtual circuits (SVCs, PVCs) to transmit fixed-size (53-byte) cells. There are no standards for ATM security, and work is in progress at the ATM Forum [23]. Next we review some of the security threats inherent in the architecture. All the cells carrying the same VPI/VCI (virtual path identifier, virtual connection identifier, respectively) are carried on the same virtual channel. Thus, eavesdropping and integrity violation attacks can be mounted to *all* the cells of a connection from a single point. In particular the cells carrying signaling information can be used to identify communicating parties. For example, capturing CONNECT or CALL PROCEEDING messages during signaling will reveal the VPI/VCIs assigned by the network to a particular connection. Flooding network with SET UP requests can be used to achieve denial-of-service attacks. Management cells can be abused to disrupt or disconnect legitimate connections. For example, by tampering with AIS/FERF cells, the attacker can cause a connection to be terminated.

3.2.1. IP over ATM. ATM networks has been deployed in high-speed backbone as the switching plane for IP traffic using IP over ATM protocols. The IP over ATM suite brings security concerns in ATM networks, many of which are similar to those in IP networks; however, their remedies are more difficult. For example, firewalls and packet filters used for access control in IP networks will require termination of ATM connection, inducing large delays and overhead. Authentication between ATMARP (ATM Address Resolution Protocol) server and hosts is a must for preventing various threads, including *spoofing, denial-of-service*, and *man-in-the-middle* attacks. For example, it is possible to send spoofed IP packets over an ATM connection, if the ATM address of an ATMARP server is known. The attacker can first establish a virtual connection to the server and then use the IP address of the victim to spoof the packets on this connection. Since the server will reply back to the attacker using the same connection the victim may not even know the attack. Similarly, the attacker can use the IP addresses of

victims to register them with the ATMARP server. Since each IP address can be used only once, the victims will be denied service.

4. TRANSPORT-LAYER SECURITY

The Secure Sockets Layer (SSL) protocol [24] was developed to ensure secure communication between the Internet browsers and servers by Netscape Corporation. Protocols such as HTTP run over SSL to provide secure connections. The Transport Layer Security (TLS) protocol [25] is expected to become the standard for secure client/server applications over the Internet. TLS v.1.0 is based on SSL v.3.0 and considered to be SSL v.3.1. Both SSL and TLS provide encryption, authentication, and integrity protection over a public network. They are composed of two subprotocols (layers): the Record Protocol and the Handshake Protocol. The *Record Protocol* is at the lowest layer and resides above a reliable transport layer protocol such as TCP. It provides encryption using symmetric cryptography (e.g., DES), and message integrity check using a keyed MAC. The *Handshake Protocol* is used to agree on cryptographic algorithms, to establish a set of keys to be used by the ciphers, and to authenticate the client. The Handshake Protocol starts with a *ClientHello* message sent to a server by a client. This message contains a random number, version information, encryption algorithms that the client supports, and a session ID. The server sends back a *ServerHello* message that also contains random data, session ID, and indicates selected cipher. In addition, the server sends a *Certificate* message that contains the server's RSA public key in an X.509 [11,12] certificate. The client verifies the server's public key, generates a 48-byte random number called the *premaster key*, encrypts it using server's public key, and sends it to the server. The client also computes a *master secret* and uses the master key to derive at a symmetric *session key*. The server performs similar operations to compute a master key and a symmetric key. After the keys are installed to the record layer, the handshake is completed. Although SSL and TLS are similar there are also important differences between them. For example, in SSL each message is transmitted with a new socket while in TLS multiple messages can be transmitted over the same socket. Security of the SSL protocol is well examined and reported to be sound, although there are some easy-to-fix problems [26]. For example, unprotected data structures (e.g., server key exchange) can be exploited to perform cryptographic attacks (e.g., *ciphersuite rollback attack* [26]).

4.1. Multicast Security

Multicasting is a group communication with single or multiple sources and multiple receivers. It has considerably more challenging security concerns than does a single source–destination communication:

1. Message authentication and confidentiality in a multicast group requires efficient key management protocols. Establishing a different key between each pair of multicast members is not scalable. Thus, most of the solutions focus on a *shared* key [27–29]. However, a single-key approach is not sufficient to authenticate the identity of a sender since it is shared by all the members. Signing each message using a public key scheme is a costly solution; thus MAC-based solutions have been proposed [30].
2. The membership dynamics (i.e., joining and leaving a multicast group) requires efficient key revocation algorithms. In particular deletion of a user from the multicast group must not reset all the keys. The solution is based on assigning multiple keys to each member and organizing the key allocation into a data structure that is easy to update [32–34]. For example, the Wallner et al. [33] key allocation scheme uses a (binary) tree structure. The group members are the leaves, and each intermediate node represents a distinct key. Each user will receive all the keys on the path to the root of the tree, and the root contains the shared group key. A group controller manages the data structure for delete and insert operations. Thus, in the key-based scheme each user gets $\log(n+1)$ keys and deletion of a user cost $2\log n - 1$ key encryptions.

5. APPLICATION-LEVEL SECURITY: KERBEROS

Kerberos is an authentication service that allows users and services to authenticate themselves to each other [31]. It is typically used when a user on a network requests a network service, and the server needs to ensure that the user is a legitimate one. For example, it enables users to log in to remote computers over the network without exposing their passwords to network packet-sniffing programs. User authentication is based on a "ticket" issued by the Kerberos key distribution center (KDC), which has two modules: authentication server (AS) and ticket-granting server (TGS). Both the user and the server are required to have keys registered with the AS. The user's key is derived from a password that is seen by only the local machine; the server key is selected randomly. The authentication between a user u and server S has the following steps:

1. The user u sends a message to the AS specifying the server S.
2. The AS produces two copies of a key called the *session key* to be used between u and S. AS encrypts one of the session keys and the identity S of the server using the user's key. Similarly, it encrypts the other session key and identity of the user with the server key. It sends both of the encrypted messages, called the "tickets" (say, m_1 and m_2, respectively) to u.
3. u can decrypt m_1 with its own key, extracting the session key and the identity of the server S. However, u cannot decrypt m_2 instead it timestamps a new message m_3 (called the *authenticator*), encrypts it with the session key and sends both m_2 and m_3 to S.
4. S decrypts m_2 with its own key to obtain the session key and the identity of user u. It then decrypts m_3

with the session key to extract the timestamp in order to authenticate the identity of the user u.

Following Step 4, all the communication between u and S will be done using the session key. However, in order to avoid performing all the steps above for each request, the TGS module in KDC issues a special ticket called the *ticket-granting ticket* (TGT). TGT behaves like a temporary password, with a lifetime of several hours only, and all other tickets are obtained using TGT.

6. WIRELESS SECURITY

Security in wireless networks is a challenging problem — the bandwidth and power limitations encourage the use of weaker cryptographic tools or keys with smaller sizes; also, the lack of point-to-point links makes it more difficult to protect the communication.

Elliptic curve crypto (ECC) systems [35] provide a remedy to some of these problems. ECC is based on discrete logarithm problem defined over the points on an elliptic curve. It is considered to be harder than the factorization problem and can provide works with much smaller key size than can other public key crypto systems [36]. Smaller key size reduces the processing overhead, and smaller digital signatures save on the bandwidth consumption.

6.1. Wireless LAN (WLAN)

WLANs use RF technology to receive and transmit data in a local-area network domain. In contrast with a wired LAN, a WLAN offers mobility and flexibility due to lack of any fixed topology. IEEE 802.11 is the most widely adapted standard for WLANs and it operates in the 2.4–2.48-GHz band.

There are several security vulnerabilities of a WLAN due to its nature: (1) any node within the transmission range of the source can eavesdrop easily, (2) unsuccessful attempts to access to a WLAN may be interpreted as a high *bit error rate* (BER) — this misinterpretation can be used to conceal an intruder's unauthorized access attack to a WLAN, and (3) the transmission medium is "shared" among the users. Thus, intentional interference (called "jamming") can be produced in a WLAN for denial of service attacks.

The *spread-spectrum* transmission technology helps countermeasure some of these problems in the WLANs. In spread spectrum, a signal is spread over the channel using two different techniques: (1) the frequency-hopping spread spectrum (FHSS), and (2) the direct-sequence spread spectrum (DSSS). An attacker must know the hopping pattern in FHSS or the codewords in DSSS to tune into the right frequency for eavesdropping. (Ironically these parameters are made public in the IEEE 802.11 standard.) Additional help comes from sophisticated network interface cards (NICs) of IEEE 802.11b devices. These cards can be equipped with a unique public and private key pair, in addition to their unique address, to prevent unauthorized access to a WLAN.

IEEE 802.11 standard provides a security capability called *wired equivalent privacy* (WEP). In WEP there is a secret 40-bit or a 128-bit key that is shared between a wireless node and an access point. Communication between a wireless station and its access point can be encrypted using the key and RSA's RC4 encryption algorithm. RC4 is a stream cipher with a variable key size and uses an *initialization vector* (IV). IV is used to produce different ciphertexts for identical plaintexts by initializing the shift registers with random bits. IV does not need to be secret but it should be unique for each transmission. However, IEEE 802.11 does not enforce the uniqueness of IV. Thus, one potential problem with the WEP is the reuse of IV, which may be exploited for cryptanalyzing and for fabricating new messages [37].

6.2. Wireless Transport Layer (WTLS)

The Wireless Transport Layer Security (WTLS) [38] protocol provides authentication, privacy and integrity for the Wireless Application Protocol (WAP) [39]. The WTLS is based on TLS v.1.0 and takes into account the characteristics of wireless world (e.g., low bandwidth, limited processing and power capacity, and connectionless datagram service). WTLS supports a rich set of cryptographic algorithms. Confidentiality is provided by using block ciphers such as DES CBC, integrity is ensured by SHA-1 [41] and MD5 [40] MAC algorithms, and the authentication is checked by RSA and Diffie–Hellman-based key exchange algorithms. WTLS does not contain any serious security problems to force an architectural change. Nevertheless there are several weak points of the protocol: (1) the computation of initialization vector (IV) is not a secret, (2) some fields in the data structures used by the protocol are not protected (one example is the sequence numbering, which enables an attacker to generate replay attacks), and (3) the key size should be at least 56 bits since 40-bit keys are not sufficient.

7. CONCLUSIONS

Heterogeneity of the Internet requires a skillful integration of the cryptographic building blocks with protocols for ensuring end-to-end security. Thus, deployment of security in the Internet cannot be confined to a particular crypto algorithm or to a particular architecture. The limitations on the processing capability or bandwidth forces sacrifices on the security (e.g., smaller key sizes, IV reuse, CRC for integrity check). Some of these problems can be addressed by efficient cryptographic tools such as ECC, and some will disappear as the technology improves. Many attacks exploit the way protocols are designed and implemented, even if these protocols may use very secure ciphers. Examples include unprotected fields in the data structures (e.g., SSL 3.0 server key exchange message) and lack of authentication in ATMARP between server and client. Finally, the security problem in the Internet degenerates to the distributed consensus problem if network elements are compromised by the adversary. For example there is no easy way to check the "correctness" of a routing exchange message if it is signed by a once-legitimate-but-compromised router.

Thus the Internet will never be absolutely secure, and creating a high cost–benefit tradeoff for the attacker, to reduce the incentive, will always remain a practical security measure.

BIOGRAPHY

Bulent Yener is an Associate Professor at the Computer Science Department at Rensselaer Polytechnic Institute. Dr. Yener received B.S. and M.S. degrees in Industrial Engineering from the Technical University of Istanbul, Turkey, and M.S. and Ph.D. degrees in Computer Science, both from Columbia University, in 1987 and 1994, respectively. He was a Member of Technical Staff at the Bell Laboratories in Murray Hill, New Jersey during 1998–2001. Before joining to the Bell Laboratories in 1998, he served as an Assistant Professor at Lehigh University and NJIT. His current research interests include quality of service in the IP networks, wireless networks, and Internet security. He has served on the Technical Program Committee of leading IEEE conferences and workshops. Dr. Yener is a member of the IEEE and serves on the editorial boards of the *Computer Networks Journal* and the *IEEE Network Magazine*. He is a member of IEEE.

BIBLIOGRAPHY

1. B. Schneier, *Applied Cryptography*, 2nd ed., Wiley, New York, 1996.
2. D. R. Stinson, *Cryptography: Theory and Practice* (*Discrete Mathematics and Its Applications*), Chapman & Hall, 1995.
3. William F. Friedman, *Index of Coincidence and Its Applications in Cryptography*, Riverbank Publication 22, Riverbank Labs., 1920, reprinted by Agean Park Press, 1976.
4. E. H. Hebern, Electronic coding machine, U.S. Patent 1,510,441,30.
5. C. E. Shannon, in N. J. A. Sloane and A. D. Wyner, eds., *Collected Papers: Claude Elmwood Shannon*, IEEE Press, New York, 1993.
6. H. Feistel, Cryptography and computer privacy, *Sci. Am.* **228**(5): 15–23 (1973).
7. M. J. Weiner, Efficient DES key search, *Proc. CRYPTO'93*, 1993.
8. J. Daemen and V. Rijmen, *Rijndael Home Page* (online) *http://www.esat.kueven.ac.be/rijmen/rijndael* (March 26, 2002).
9. W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Trans. Inform. Theory* **IT-22**: 644–654 (1976).
10. R. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM* **21**(2): 120–126 (1978).
11. ITU-T Recommendation X.509, *Information Technology — Open System Interconnection — The Directory: Authentication Framework*, 1997.
12. R. Housley, W. Ford, W. Polk, and D. Solo, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, IETF RFC 2459, 1999.
13. J. Postel and J. Reynolds, File Transfer Protocol, *IETF RFC 959*, 1985.
14. J. Postel, *Transmission Control Protocol*, IETF RFC 791, 19.
15. S. M. Bellovin, Security Problems in the TCP/IP Protocol Suite, *ACM Comput. Commun. Rev.* **19**(2): 32–48 (1989).
16. S. M. Bellovin and W. R. Cheswick, *Firewalls and Internet Security*, Addison-Wesley, New York, 1994.
17. T. Ylonen et al., *SSH Protocol Architecture* (online) *draft-ietfsecsh-architecture-12.txt*, 2002.
18. S. Kent, C. Lynn, and K. Seo, Secure Border Gateway Protocol (S-BGP), *IEEE JSAC Network Security* **18**(34): 582–592 (2000).
19. M. Raynal, *Distributed Algorithms and Protocols*, Wiley, New York, 1988.
20. S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*. IETF RFC 2401, 1998.
21. S. Bellovin, Problem areas for the IP security protocols, *Proc. 6th USENIX Security Symp.* 1996, pp. 205–214.
22. D. Harkins and D. Carrel, *The Internet Key Exchange*, IETF RFC 2409, 1998.
23. ATM Forum. *http://www.atmforum.org.*
24. A. Frier, P. Karlton, and P. Koccher, *The SSL3.0 Protocol Version 3.0,* Netscape, 1996 (online) *http://home.netscape.com/eng/ssl3/* (March 26, 2002).
25. T. Dierks and C. Allen, *The TLS Protocol Version 1.0*, IETF RFC 2246, 1999.
26. D. Wagner and B. Schneier, Analysis of the SSL 3.0 protocol, *Proc. 2nd USENIX Workshop on Electronic Commerce*, USENIX Press, 1996, pp. 29–40 (online) *http://citeseer.nj.nec.com/article/wagner96analysi.html* (March 26, 2002).
27. H. Harney and C. Muckenhirn, *Group Key Management Protocol* (*GKMP*) *Specification*, IETF RFC 2093, 1997.
28. H. Harney and C. Muckenhirn, *Group Key Management Protocol* (*GKMP*) *Architecture*, IETF RFC 2094, 1997.
29. S. Mittra, Iolus: A framework for scalable secure multicasting, *Proc. ACM SIGCOMM'97*, 1997.
30. R. Canetti et al., Multicast security: A taxonomy and efficient constructions, *Proc. IEEE INFOCOM'99*, 1999.
31. J. Kohl and C. Neuman, *The Kerberos Network Authentication Service* (*V5*), IETF RFC 1510, 1993.
32. A. Fiat and M. Naor, Broadcast encryption, *Advances in Cryptography — Crypto'92*, 1995, Vol. 8, pp. 189–200.
33. D. M. Wallner, E. J. Harder, and R. C. Agee, *Key Management for Multicast: Issues and Architectures*, IETF RFC 2627, 1999.
34. C. K. Wong and S. Lam, Digital signature for flows and multicasts, *Proc. IEEE ICNP'98'*, 1998.
35. N. Koblitz, Elliptic curve cryptosystems, *Math. Comput.* **48**: 203–209 (1987).
36. Certicom White Paper, *Current Public-Key Cryptographic Systems*, 1997 (online) *http://www.certicom.com* (March 26, 2002).
37. N. Borisov, I. Goldberg, and D. Wagner, Intercepting mobile communications: The insecurity of 802.11, *Proc. Mobile Computing and Networking*, 2001.

38. WAP Forum, *Wireless Application Protocol — Wireless Transport Layer Security Specification version 1* (online) *http://www.wapforum.org* (March 26, 2002).

39. WAP Forum, *Wireless Application Protocol* (online) *http://www.wapforum.org* (March 26, 2002).

40. R. Rivest, *The MD5 Message-Digest Algorithm*, IETF RFC 1321, 1992.

41. Federal Information Processing Standard Publication 180-1, 1995 (online) *http://www.itl.nist.gov/fibspubs/fip180-1.htm* (March 26, 2002).