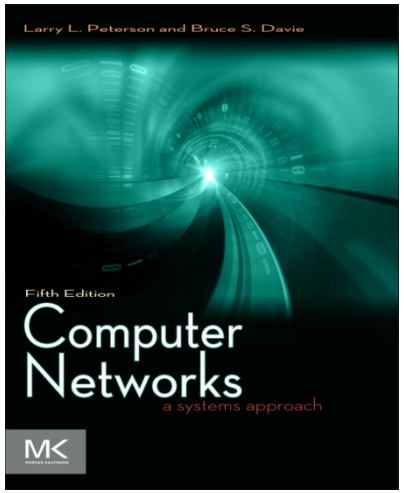


09 Advanced Topics (Chapter 8)

Network Security

Problem

- Computer networks are typically a shared resource used by many applications representing different interests.
- The Internet is particularly widely shared, being used by competing businesses, mutually antagonistic governments, and opportunistic criminals.
- Unless security measures are taken, a network conversation or a distributed application may be compromised by an adversary.

Problem

- Consider some threats to secure use of, for example, the World Wide Web.
  - Suppose you are a customer using a credit card to order an item from a website.
    - An obvious threat is that an adversary would eavesdrop on your network communication, reading your messages to obtain your credit card information.
    - It is possible and practical, however, to encrypt messages so as to prevent an adversary from understanding the message contents. A protocol that does so is said to provide *confidentiality.*
    - Taking the concept a step farther, concealing the quantity or destination of communication is called *traffic confidentiality*

Problem

- Even with confidentiality there still remain threats for the website customer.
  - An adversary who can't read the contents of your encrypted message might still be able to change a few bits in it, resulting in a valid order for, say, a completely different item or perhaps 1000 units of the item.
  - There are techniques to detect, if not prevent, such tampering.
  - A protocol that detects such message tampering provides *data integrity.*
  - The adversary could alternatively transmit an extra copy of your message in a *replay attack.*

Problem

- To the website, it would appear as though you had simply ordered another of the same item you ordered the first time.
  - A protocol that detects replays provides *originality.*
  - Originality would not, however, preclude the adversary intercepting your order, waiting a while, then transmitting it—in effect, delaying your order.
  - The adversary could thereby arrange for the item to arrive on your doorstep while you are away on vacation, when it can be easily snatched. A protocol that detects such delaying tactics is said to provide *timeliness.*
- Data integrity, originality, and timeliness are considered aspects of the more general property of integrity.

Problem

- Another threat to the customer is unknowingly being directed to a false website.
  - This can result from a DNS attack, in which false information is entered in a Domain Name Server or the name service cache of the customer's computer.
  - This leads to translating a correct URL into an incorrect IP address—the address of a false website.
  - A protocol that ensures that you really are talking to whom you think you're talking is said to provide *authentication.*
  - *Authentication entails integrity since it is meaningless* to say that a message came from a certain participant if it is no longer the same message.

Problem

- The owner of the website can be attacked as well. Some websites have been defaced; the files that make up the website content have been remotely accessed and modified without authorization.
- That is an issue of *access control: enforcing the rules* regarding who is allowed to do what. Websites have also been subject to Denial of Service (DoS) attacks, during which would-be customers are unable to access the website because it is being overwhelmed by bogus requests.
- Ensuring a degree of access is called *availability.*

Problem

- In addition to these issues, the Internet has notably been used as a means for deploying malicious code that exploits vulnerabilities in end-systems.
- *Worms, pieces of* self-replicating code that spread over networks, have been known for several decades and continue to cause problems, as do their relatives, *viruses, which are spread by the* transmission of "infected" files.
- Infected machines can then be arranged into *botnets* which can be used to inflict further harm, such as launching DoS attacks.

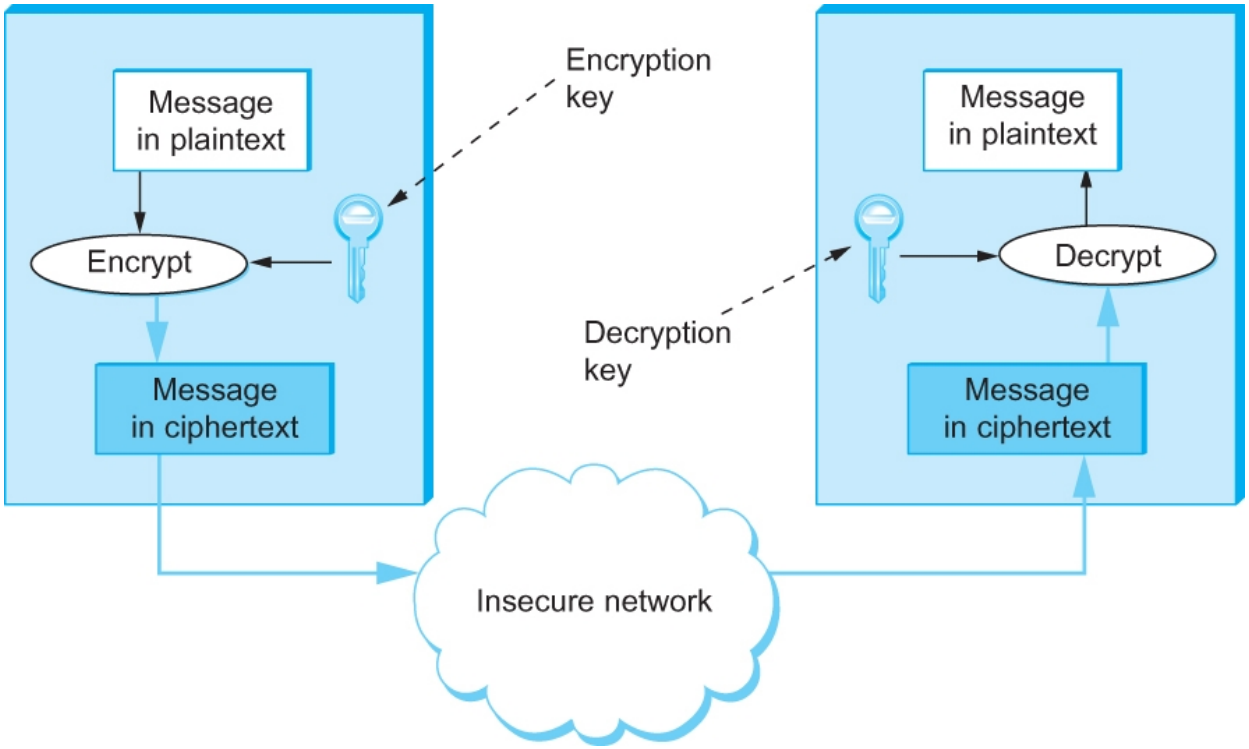
Chapter Outline

- Cryptographic Building Blocks
- Key Pre Distribution
- Authentication Protocols
- Example Systems
- Firewalls

Cryptograhic Building Blocks

- We introduce the concepts of cryptography-based security step by step.
- The first step is the cryptographic algorithms—ciphers and cryptographic hashes
- Cryptographic algorithms are parameterized by *keys*

Cryptograhic Building Blocks



Symmetric-key encryption and decryption

Cryptograhic Building Blocks

- Principles of Ciphers
  - Encryption transforms a message in such a way that it becomes unintelligible to any party that does not have the secret of how to reverse the transformation.
  - The sender applies an *encryption function to the original plaintext message, resulting in a ciphertext* message that is sent over the network.
  - The receiver applies a secret *decryption function–the inverse of the encryption function–to recover the original* plaintext.

Cryptograhic Building Blocks

- Principles of Ciphers
  - The ciphertext transmitted across the network is unintelligible to any eavesdropper, assuming she doesn't know the decryption function.
  - The transformation represented by an encryption function and its corresponding decryption function is called a *cipher.*
  - The basic requirement for an encryption algorithm is that it turn plaintext into ciphertext in such a way that only the intended recipient—the holder of the decryption key—can recover the plaintext.

Cryptograhic Building Blocks

- Principles of Ciphers
  - It is important to realize that when a potential attacker receives a piece of ciphertext, he may have more information at his disposal than just the ciphertext itself.
  - Known plaintext attack
  - Ciphetext only attack
  - Chosen plaintext attack

Cryptograhic Building Blocks

- Principles of Ciphers
  - Most ciphers are *block ciphers: they are defined to take as input a plaintext block* of a certain fixed size, typically 64 to 128 bits.
  - Using a block cipher to encrypt each block independently—known as *electronic codebook (ECB) mode encryption—has the* weakness that a given plaintext block value will always result in the same ciphertext block.
  - Hence recurring block values in the plaintext are recognizable as such in the ciphertext, making it much easier for a cryptanalyst to break the cipher.

Cryptograhic Building Blocks

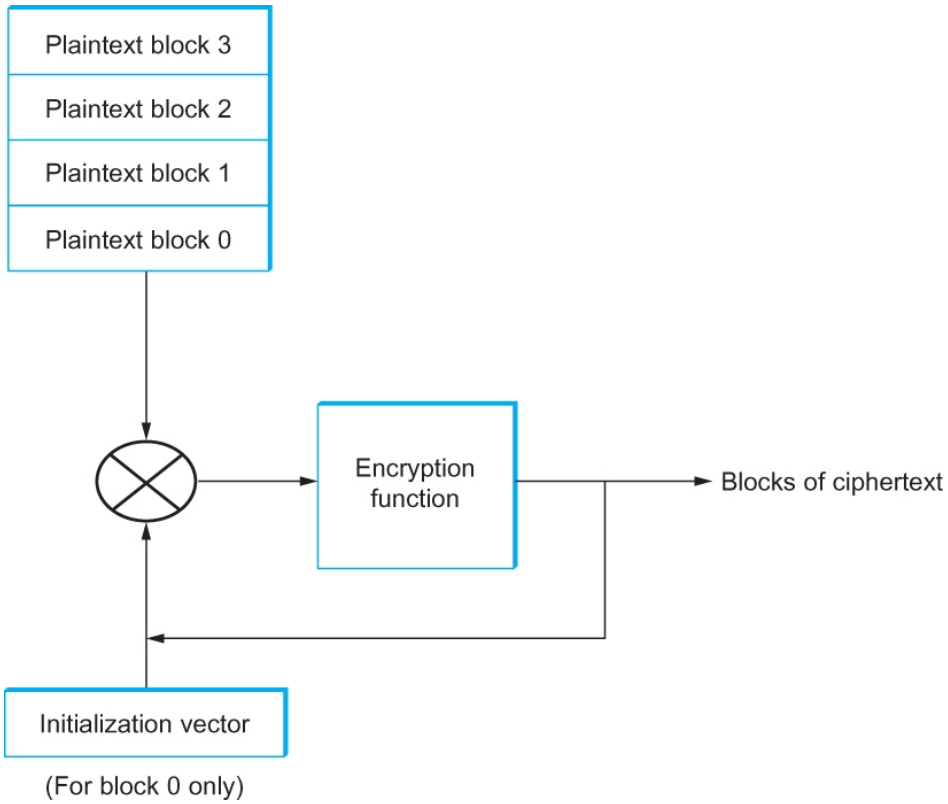
- Principles of Ciphers
  - Block ciphers are always augmented to make the ciphertext for a block vary depending on context. Ways in which a block cipher may be augmented are called *modes of operation.*

Cryptograhic Building Blocks

- Block Ciphers
  - *A common mode of operation is cipher block chaining* (CBC), in which each plaintext block is XORed with the previous block's ciphertext before being encrypted.
    - The result is that each block's ciphertext depends in part on the preceding blocks, i.e. on its context. Since the first plaintext block has no preceding block, it is XORed with a random number.
      - That random number, called an *initialization vector (IV), is included with the series of ciphertext blocks so that the first ciphertext* block can be decrypted.

Cryptograhic Building Blocks

- Block Ciphers



Cipher block chaining (CBC).

Cryptograhic Building Blocks

- Symmetric Key Ciphers
  - In a symmetric-key cipher, both participants in a communication share the same key. In other words, if a message is encrypted using a particular key, the same key is required for decrypting the message.

Cryptograhic Building Blocks

- Symmetric Key Ciphers
  - The U.S. National Institute of Standards and Technology (NIST) has issued standards for a series of symmetric-key ciphers.
  - *Data Encryption Standard (DES) was the* first, and it has stood the test of time in that no cryptanalytic attack better than brute force search has been discovered.
  - Brute force search, however, has gotten faster. DES's keys (56 independent bits) are now too small given current processor speeds.

Cryptograhic Building Blocks

- Symmetric Key Ciphers
  - NIST also standardized the cipher *Triple DES (3DES), which leverages the cryptanalysis resistance* of DES while in effect increasing the key size.
  - A 3DES key has 168 (= 3256) independent bits, and is used as three DES keys;
    - let's call them DES-key1, DES-key2, and DES-key3.
    - 3DES-encryption of a block is performed by first DES-encrypting the block using DES-key1, then DES-*decrypting the result using DES-key2, and finally* DES-encrypting that result using DES-key3.
    - Decryption involves decrypting using DES-key3, then encrypting using DES-key2, then decrypting using DES-key1

Cryptograhic Building Blocks

- Symmetric Key Ciphers
  - 3DES is being superseded by the *Advanced Encryption Standard (AES) standard* issued by NIST in 2001.
  - The cipher selected to become that standard (with a few minor modifications) was originally named Rijndael (pronounced roughly like “Rhine dahl”) based on the names of its inventors, Daemen and Rijmen.
  - AES supports key lengths of 128, 192, or 256 bits, and the block length is 128 bits.

Cryptograhic Building Blocks

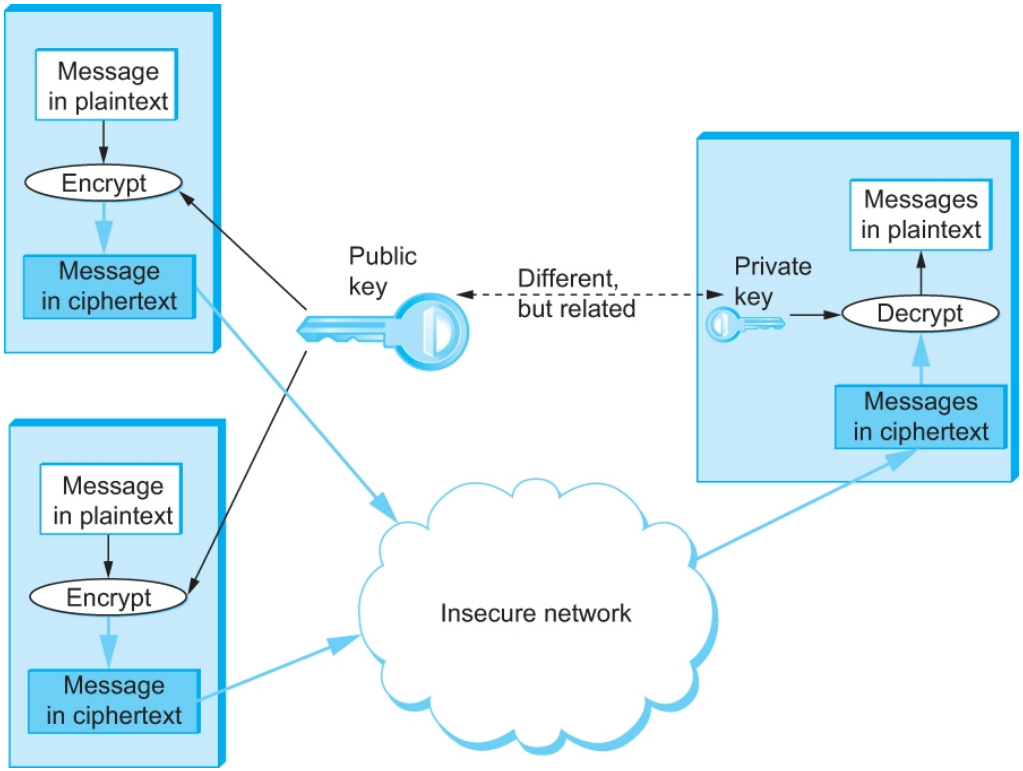
- Public Key Ciphers
  - An alternative to symmetric-key ciphers is asymmetric, or public-key, ciphers.
  - Instead of a single key shared by two participants, a public-key cipher uses a pair of related keys, one for encryption and a different one for decryption.
  - The pair of keys is “owned” by just one participant.
  - The owner keeps the decryption key secret so that only the owner can decrypt messages; that key is called the *private key.*

Cryptograhic Building Blocks

- Public Key Ciphers
  - The owner makes *the* encryption key public, so that anyone can encrypt messages for the owner; that key is called the *public key.*
  - Obviously, for such a scheme to work it must not be possible to deduce the private key from the public key.
  - Consequently any participant can get the public key and send an encrypted message to the owner of the keys, and only the owner has the private key necessary to decrypt it.

Cryptograhic Building Blocks

- Public Key Ciphers



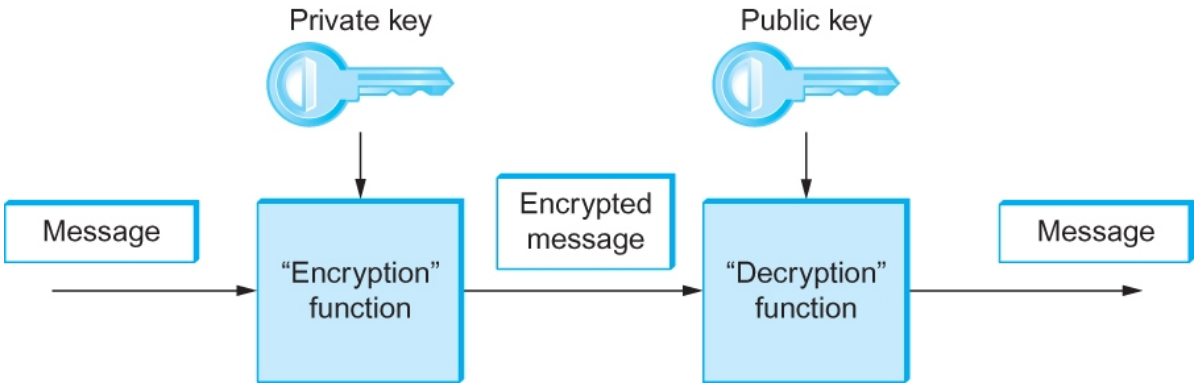
Public-key encryption

Cryptograhic Building Blocks

- Public Key Ciphers
  - An important additional property of public-key ciphers is that the private “decryption” key can be used with the encryption algorithm to encrypt messages so that they can only be decrypted using the public “encryption” key.
  - This property clearly wouldn’t be useful for confidentiality since anyone with the public key could decrypt such a message.
  - This property is, however, useful for authentication since it tells the receiver of such a message that it could only have been created by the owner of the keys.

Cryptograhic Building Blocks

- Public Key Ciphers



Authentication using public keys

Cryptograhic Building Blocks

- Public Key Ciphers
  - The concept of public-key ciphers was first published in 1976 by Diffie and Hellman.
  - The best-known public-key cipher is RSA, named after its inventors: Rivest, Shamir, and Adleman.
    - RSA relies on the high computational cost of factoring large numbers.
  - Another public-key cipher is ElGamal.
    - Like RSA, it relies on a mathematical problem, the discrete logarithm problem, for which no efficient solution has been found, and requires keys of at least 1024 bits.

Cryptograhic Building Blocks

- Authenticator
  - An *authenticator is a value, to be included in a transmitted message, that can be* used to verify simultaneously the authenticity and the data integrity of a message.
  - One kind of authenticator combines encryption and a *cryptographic hash function.*
    - Cryptographic hash algorithms are treated as public knowledge, as with cipher algorithms.
    - A cryptographic hash function (also known as a cryptographic checksum) is a function that outputs sufficient redundant information about a message to expose any tampering.

Cryptograhic Building Blocks

- Authenticator
  - Just as a checksum or CRC exposes bit errors introduced by noisy links, a cryptographic checksum is designed to expose deliberate corruption of messages by an adversary.
    - The value it outputs is called a *message digest and,* like an ordinary *checksum,* is appended to the message.
    - All the message digests produced by a given hash have the same number of bits regardless of the length of the original message.
    - Since the space of possible input messages is larger than the space of possible message digests, there will be different input messages that produce the same message digest, like collisions in a hash table.

Cryptograhic Building Blocks

- Authenticator
  - There are several common cryptographic hash algorithms, including MD5 (for Message Digest 5) and Secure Hash Algorithm 1 (SHA-1). MD5 outputs a 128-bit digest, and SHA-1 outputs a 160-bit digest
  - A digest encrypted with a public key algorithm but using the private key is called a *digital signature* because it provides nonrepudiation like a written signature.

Cryptograhic Building Blocks

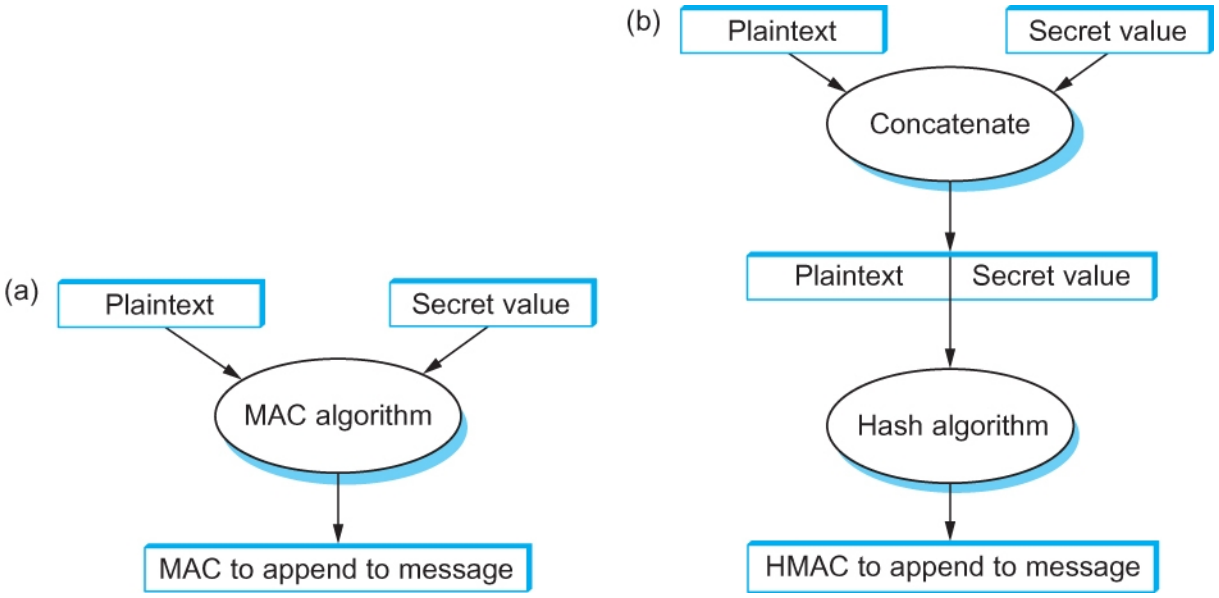
- Authenticator
  - Another kind of authenticator is similar, but instead of encrypting a hash, it uses a hash-like function that takes a secret value (known to only the sender and the receiver) as a parameter.
  - Such a function outputs an authenticator called a *message authentication code (MAC).*
  - The sender appends the MAC to her plaintext message.
  - The receiver recomputes the MAC using the plaintext and the secret value, and compares that recomputed MAC to the received MAC.

Cryptograhic Building Blocks

- Authenticator
  - A common variation on MACs is to apply a cryptographic hash (such as MD5 or SHA-1) to the concatenation of the plaintext message and the secret value.
  - The resulting digest is called a *hashed message authentication code (HMAC)* since it is essentially a MAC.
  - The HMAC, but not the secret value, is appended to the plaintext message.
  - Only a receiver who knows the secret value can compute the correct HMAC to compare with the received HMAC.

Cryptograhic Building Blocks

- Authenticator



Computing a MAC versus computing an HMAC

Key Pre Distribution

- To use ciphers and authenticators, the communicating participants need to know what keys to use.
- In the case of a symmetric-key cipher, how does a pair of participants obtain the key they share?
- In the case of a public-key cipher, how do participants know what public key belongs to a certain participant?
- The answer differs depending on whether the keys are short-lived *session keys or longer-lived pre-distributed keys.*

Key Pre Distribution

- A session key is a key used to secure a single, relatively short episode of communication: a session.
  - Each distinct session between a pair of participants uses a new session key, which is always a symmetric-key key for speed.
  - The participants determine what session key to use by means of a protocol—a session key establishment protocol.
  - A session key establishment protocol needs its own security (so that, for example, an adversary cannot learn the new session key); that security is based on the longer-lived pre-distributed keys.

Key Pre Distribution

- There are several motivations for this division of labor between session keys and pre-distributed keys:
  - Limiting the amount of time a key is used results in less time for computationally intensive attacks, less ciphertext for cryptanalysis, and less information exposed should the key be broken.
  - Pre-distribution of symmetric keys is problematic.
  - Public key ciphers are generally superior for authentication and session key establishment but too slow to use encrypting entire messages for confidentiality.

Key Pre Distribution

- Pre-Distribution of Public Keys
  - The algorithms to generate a matched pair of public and private keys are publicly known, and software that does it is widely available.
  - So if Alice wanted to use a public key cipher, she could generate her own pair of public and private keys, keep the private key hidden, and publicize the public key.
  - But how can she publicize her public key— assert that it belongs to her—in such a way that other participants can be sure it really belongs to her?

Key Pre Distribution

- Pre-Distribution of Public Keys
  - A complete scheme for certifying bindings between public keys and identities— what key belongs to who —is called a Public Key Infrastructure (PKI).
  - A PKI starts with the ability to verify identities and bind them to keys out of band. By “out of band,” we mean something outside the network and the computers that comprise it, such as in the following scenarios.
  - If Alice and Bob are individuals who know each other, then they could get together in the same room and Alice could give her public key to Bob directly, perhaps on a business card.

Key Pre Distribution

- Pre-Distribution of Public Keys
  - If Bob is an organization, Alice the individual could present conventional identification, perhaps involving a photograph or fingerprints.
  - If Alice and Bob are computers owned by the same company, then a system administrator could configure Bob with Alice's public key.
  - A digitally signed statement of a public key binding is called a *public key certificate, or simply a certificate*

Key Pre Distribution

- Pre-Distribution of Public Keys
  - One of the major standards for certificates is known as X.509. This standard leaves a lot of details open, but specifies a basic structure. A certificate clearly must include
    - the identity of the entity being certified
    - the public key of the entity being certified
    - the identity of the signer
    - the digital signature
    - a digital signature algorithm identifier (which cryptographic hash and which cipher)

Key Pre Distribution

- Pre-Distribution of Public Keys
  - Certification Authorities
    - A *certification authority or certificate authority (CA) is an entity claimed (by someone)* to be trustworthy for verifying identities and issuing public key certificates.
    - There are commercial CAs, governmental CAs, and even free CAs.
    - To use a CA, you must know its own key. You can learn that CA's key, however, if you can obtain a chain of CA-signed certificates that starts with a CA whose key you already know.
    - Then you can believe any certificate signed by that new CA

Structure of X.509 digital certificate

- Certificate
  - Version Number
  - Serial Number
  - Signature Algorithm ID
  - Issuer Name
  - Validity period
    - Not Before
    - Not After
  - Subject name
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (optional)
  - Subject Unique Identifier (optional)
  - Extensions (optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

Key Pre Distribution

- Web of Trust Model
  - There is no central certifying authority
  - Mutual trust among the nodes
  - An authority provides a level of confidence on the certificate
  - A node might have to collect several certificates to gain trust of others

Key Pre Distribution

- Certificate revocation
  - What to do when an issued certificate is being misused?
  - Create an Certificate Revocation List (CRL) and publish
  - Every certificate has an expiry date after which it can be removed from CRL

Key Pre Distribution

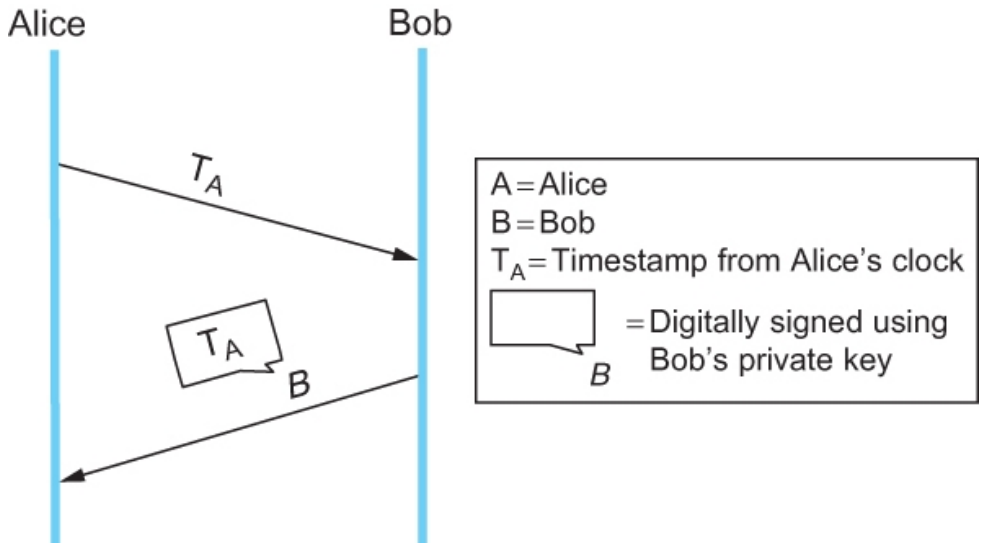
- Pre-Distribution of Symmetric Keys
  - If Alice wants to use a secret-key cipher to communicate with Bob, she can't just pick a key and send it to to him because, without already having a key, they can't encrypt this key to keep it confidential and they can't authenticate each other.
  - As with public keys, some pre-distribution scheme is needed.
  - Pre-distribution is harder for symmetric keys than for public keys for two obvious reasons:
    - While only one public key per entity is sufficient for authentication and confidentiality, there must be a symmetric key for each pair of entities who wish to communicate. If there are N entities, that means $N(N-1)/2$ keys.
    - Unlike public keys, secret keys must be kept secret.
  - Solution
    - Use of a Key Distribution Center (KDC)
    - The KDC shares a key with each entity that can be used to authenticate two participating entities who want a mutual communication

Key Pre-distribution

- Use of Timestamp
  - Requires distributed clock synchronization
  - Clock synchronization itself has to be protected
- Use of nonce (a random number used only once)
  - Requires to keep track of all the past nonces
- Combine timestamp and nonce
- Challenge response protocol

Key Pre Distribution

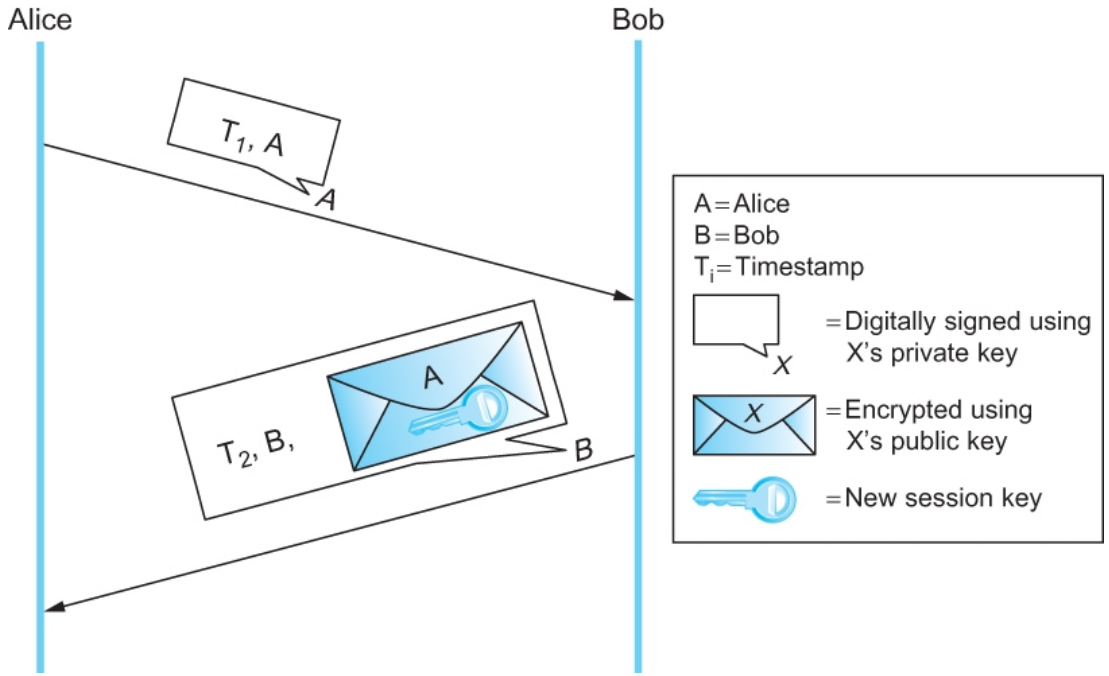
- Pre-Distribution of Symmetric Keys
  - Challenge response Protocols



A challenge-response protocol

Key Pre Distribution

- Pre-Distribution of Symmetric Keys
  - Public Key Authentication Protocols

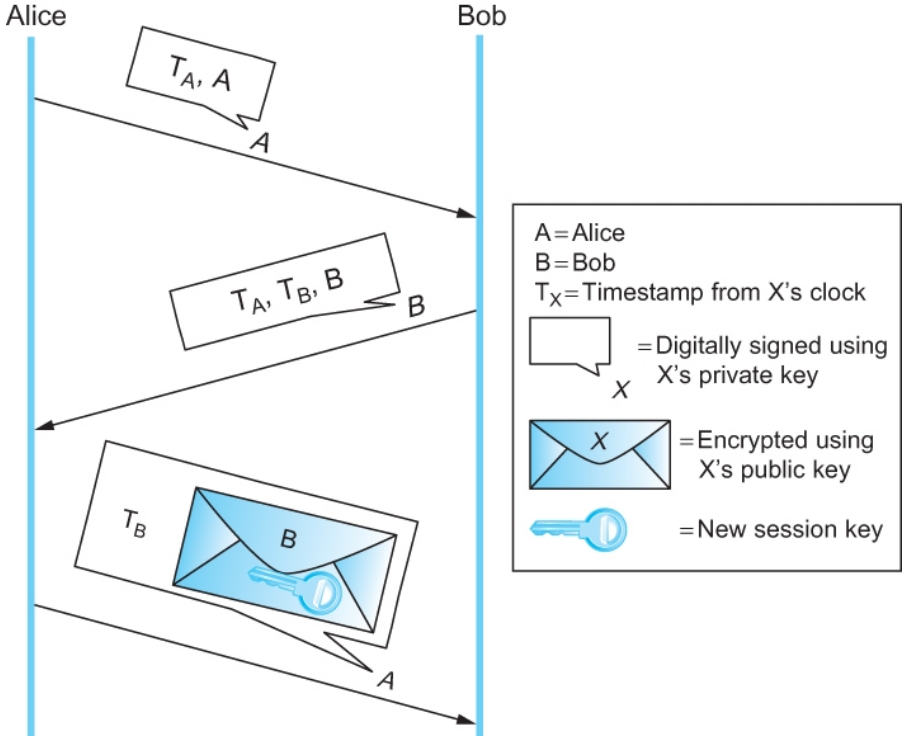


A public-key authentication protocol that depends on synchronization

Still Synchronization remains an issue that can be partially handled with nonces

Key Pre Distribution

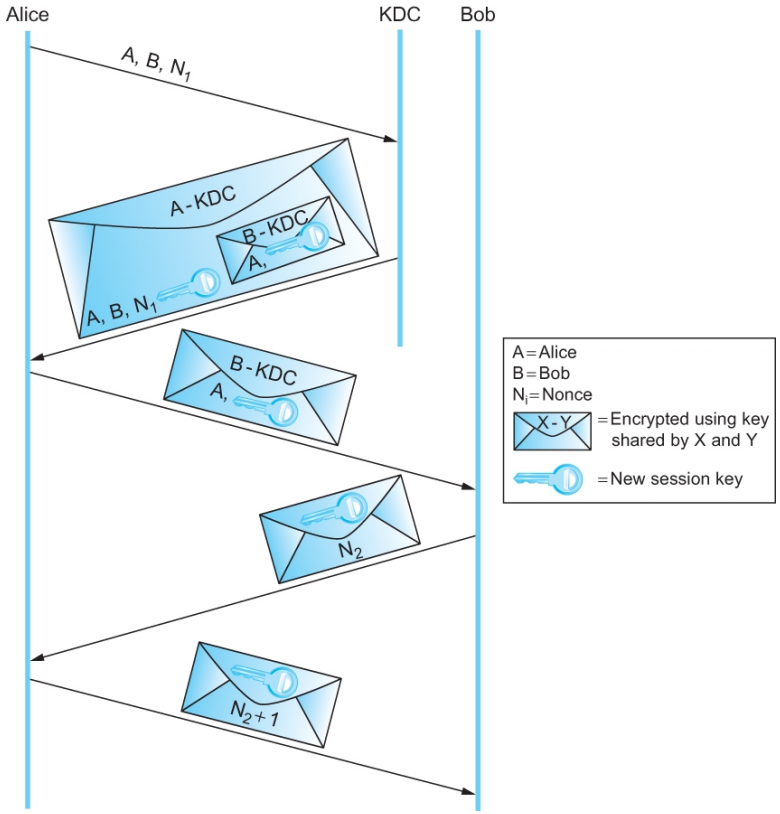
- Pre-Distribution of Symmetric Keys
  - Public Key Authentication Protocols



A public-key authentication protocol that does not depend on synchronization. Alice checks her own timestamp against her own clock, and likewise for Bob.

Key Pre Distribution

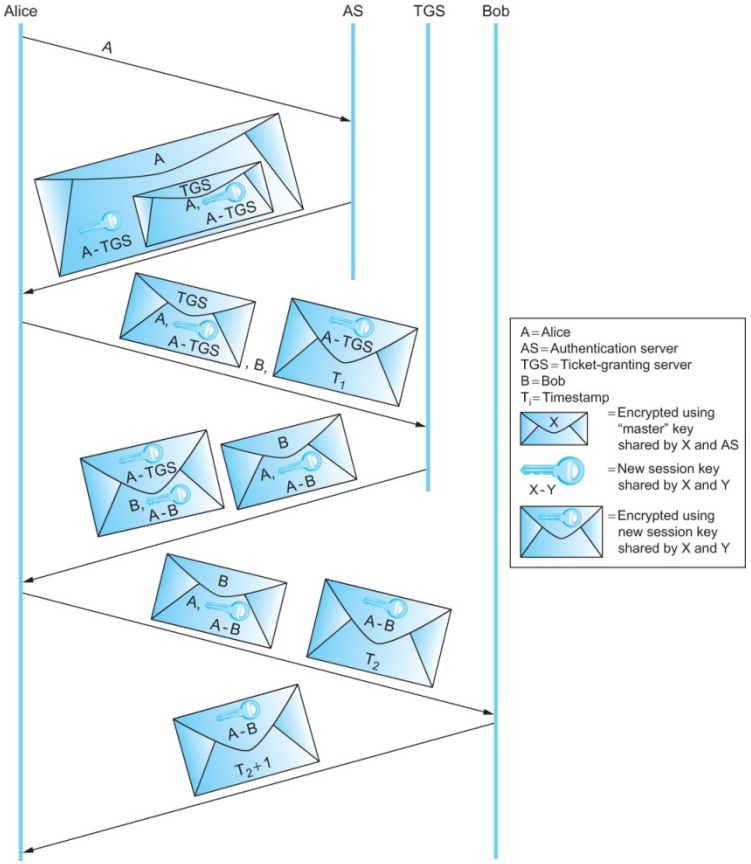
- Pre-Distribution of Symmetric Keys
  - Symmetric Key Authentication Protocols



The Needham-Schroeder authentication protocol

Key Pre Distribution

- Pre-Distribution of Symmetric Keys
  - Symmetric Key Authentication Protocols



Kerberos Authentication

Key Pre Distribution

- Pre-Distribution of Symmetric Keys
  - Diffie-Hellman Key Agreement
    - The Diffie-Hellman key agreement protocol establishes a session key without using any pre-distributed keys.
    - The messages exchanged between Alice and Bob can be read by anyone able to eavesdrop, and yet the eavesdropper won't know the session key that Alice and Bob end up with.
    - On the other hand, Diffie-Hellman doesn't authenticate the participants.
    - Since it is rarely useful to communicate securely without being sure whom you're communicating with, Diffie-Hellman is usually augmented in some way to provide authentication.
    - One of the main uses of Diffie-Hellman is in the Internet Key Exchange (IKE) protocol, a central part of the IP Security (IPSEC) architecture

Key Pre Distribution

- Pre-Distribution of Symmetric Keys
  - Diffie-Hellman Key Agreement
    - The Diffie-Hellman protocol has two parameters, p and g, both of which are public and may be used by all the users in a particular system.
    - Parameter p must be a prime number. The integers mod p (short for modulo p) are 0 through p − 1, since x mod p is the remainder after x is divided by p, and form what mathematicians call a *group* under multiplication.
    - Parameter g (usually called a generator) must be a *primitive root* of p: for every number n from 1 through p − 1 there must be some value k such that $n = g^k \bmod p$.

Key Pre Distribution

- Pre-Distribution of Symmetric Keys
  - Diffie-Hellman Key Agreement
    - For example, if p were the prime number 5 (a real system would use a much larger number), then we might choose 2 to be the generator g since:

$$1 = 2^0 \bmod p$$

$$2 = 2^1 \bmod p$$

$$3 = 2^3 \bmod p$$

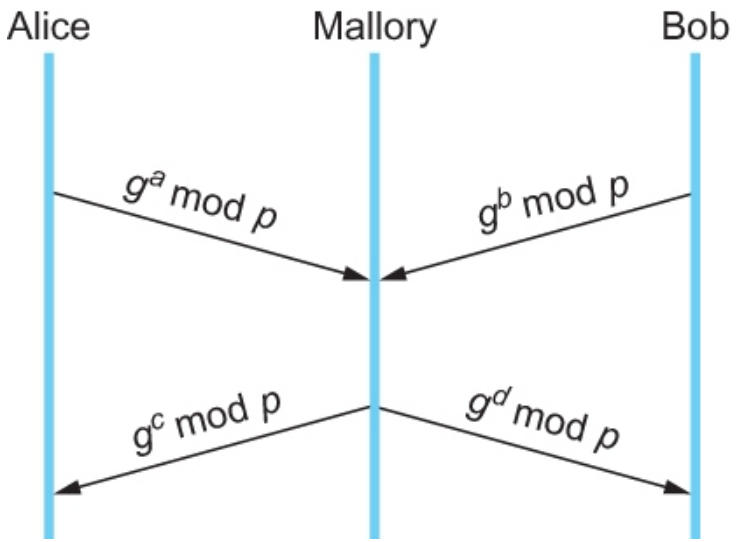
$$4 = 2^2 \bmod p$$

Key Pre Distribution

- Pre-Distribution of Symmetric Keys
  - Diffie-Hellman Key Agreement
    - Suppose Alice and Bob want to agree on a shared symmetric key. Alice and Bob, and everyone else, already know the values of p and g.
    - Alice generates a random private value a and Bob generates a random private value b.
    - Both a and b are drawn from the set of integers {1, ..., p−1}.
    - Alice and Bob derive their corresponding public values—the values they will send to each other unencrypted—as follows.
    - Alice's public value is $g^a \bmod p$
    - and Bob's public value is $g^b \bmod p$
    - They then exchange their public values. Finally, Alice computes
    - $g^{ab} \bmod p = (g^b \bmod p)^a \bmod p$
    - and Bob computes
    - $g^{ba} \bmod p = (g^a \bmod p)^b \bmod p$.

Key Pre Distribution

- Pre-Distribution of Symmetric Keys



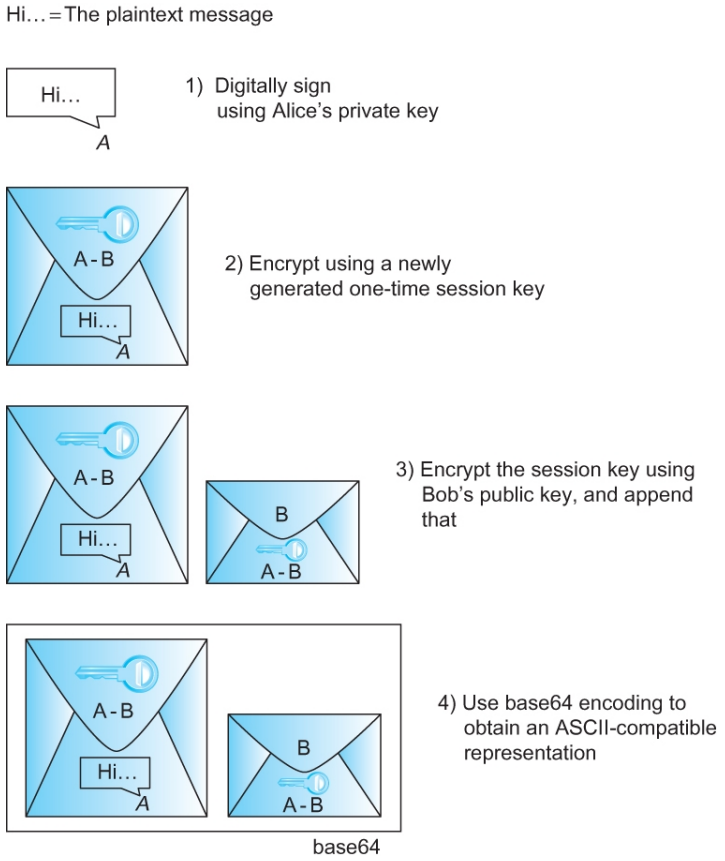
A man-in-the-middle attack

Example Systems

- Pretty Good Privacy (PGP)
  - Pretty Good Privacy (PGP) is a widely used approach to providing security for electronic mail. It provides authentication, confidentiality, data integrity, and nonrepudiation.
  - Originally devised by Phil Zimmerman, it has evolved into an IETF standard known as OpenPGP
  - PGP's confidentiality and receiver authentication depend on the receiver of an email message having a public key that is known to the sender.
  - To provide sender authentication and nonrepudiation, the sender must have a public key that is known by the receiver.
  - These public keys are pre-distributed using certificates and a web-of-trust PKI.
  - PGP supports RSA and DSS for public key certificates.

Example Systems

- Pretty Good Privacy (PGP)



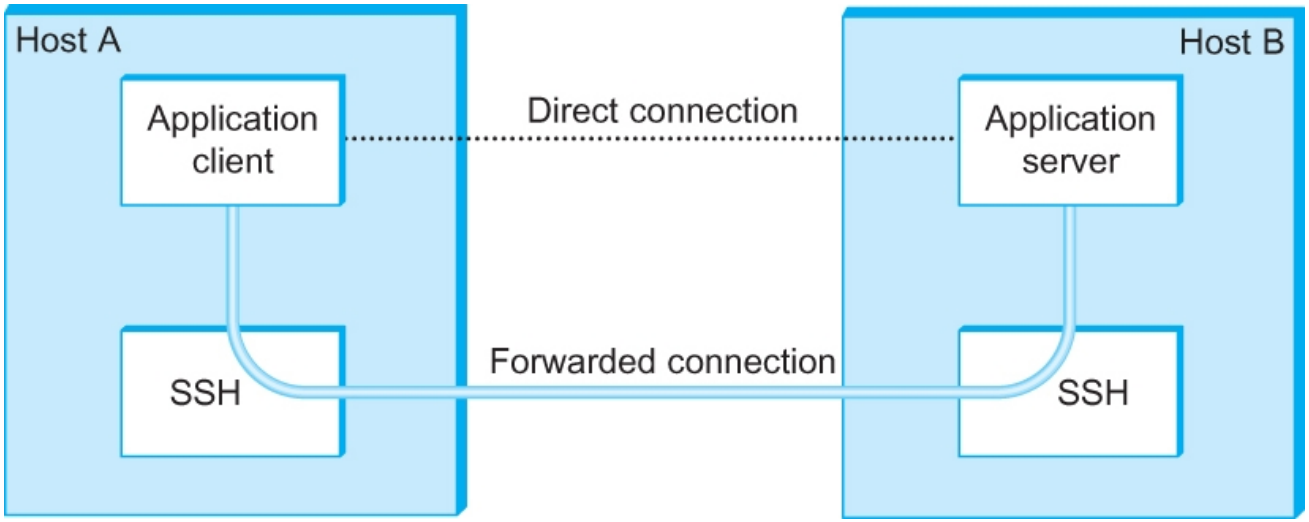
PGP's steps to prepare a message for emailing from Alice to Bob

Example Systems

- Secure Shell (SSH)
  - The Secure Shell (SSH) protocol is used to provide a remote login service, and is intended to replace the less-secure Telnet and rlogin programs used in the early days of the Internet.
  - SSH is most often used to provide strong client/server authentication/ message integrity—where the SSH client runs on the user's desktop machine and the SSH server runs on some remote machine that the user wants to log into—but it also supports confidentiality.
  - Telnet and rlogin provide none of these capabilities.
  - Note that "SSH" is often used to refer to both the SSH protocol and applications that use it; you need to figure out which from the context.

Example Systems

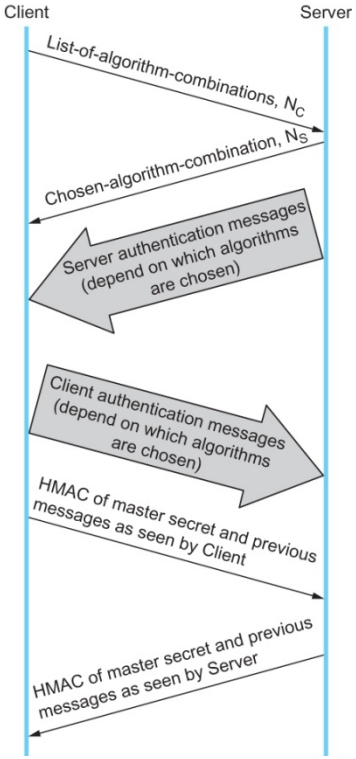
- Secure Shell (SSH)



Using SSH port forwarding to secure other TCP-based applications

Example Systems

- Transport Layer Security (TLS, SSL, HTTPS)



Handshake protocol to establish TLS session

Example Systems

- IP Security (IPSec)
  - Support for IPsec, as the architecture is called, is optional in IPv4 but mandatory in IPv6.
  - IPsec is really a framework (as opposed to a single protocol or system) for providing all the security services discussed throughout this chapter.
  - IPsec provides three degrees of freedom.
    - First, it is highly modular, allowing users (or more likely, system administrators) to select from a variety of cryptographic algorithms and specialized security protocols.
    - Second, IPsec allows users to select from a large menu of security properties, including access control, integrity, authentication, originality, and confidentiality.
    - Third, IPsec can be used to protect “narrow” streams (e.g., packets belonging to a particular TCP connection being sent between a pair of hosts) or “wide” streams (e.g., all packets flowing between a pair of routers).

Example Systems

- IP Security (IPSec)
  - When viewed from a high level, IPsec consists of two parts.
  - The first part is a pair of protocols that implement the available security services.
    - They are the Authentication Header (AH), which provides access control, connectionless message integrity, authentication, and antireplay protection, and the Encapsulating Security Payload (ESP), which supports these same services, plus confidentiality.
    - AH is rarely used so we focus on ESP here.
  - The second part is support for key management, which fits under an umbrella protocol known as ISAKMP:
    - Internet Security Association and Key Management Protocol.

Example Systems

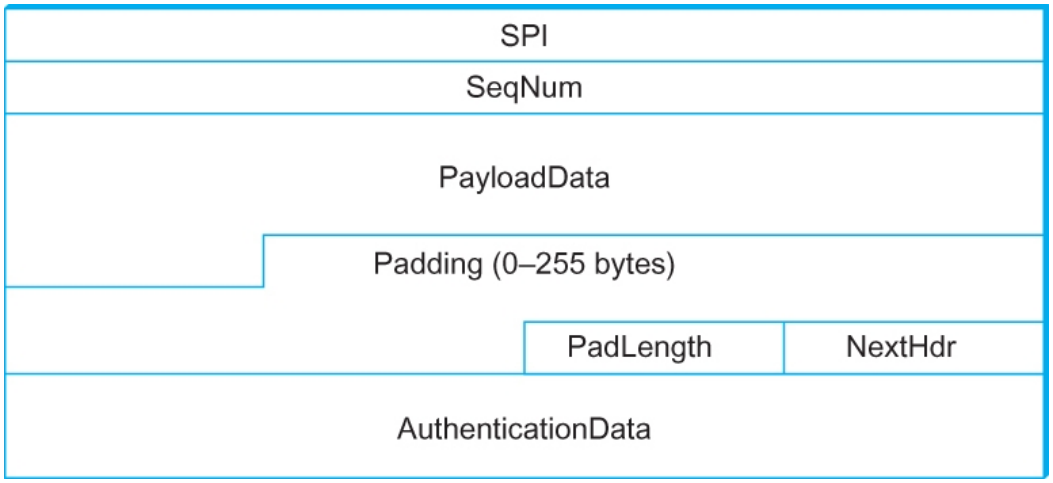
- IP Security (IPSec)
  - The abstraction that binds these two pieces together is the *security association (SA).*
  - *An SA is a simplex (one-way) connection with one or more of the available* security properties.
  - Securing a bidirectional communication between a pair of hosts — corresponding to a TCP connection, for example—requires two SAs, one in each direction.
  - Although IP is a connectionless protocol, security depends on connection state information such as keys and sequence numbers.
  - When created, an SA is assigned an ID number called a *security parameters index (SPI) by the receiving machine*

Example Systems

- IP Security (IPSec)
  - IPsec supports a *tunnel mode as well as the more straightforward transport mode.*
  - Each SA operates in one or the other mode.
  - In a transport mode SA, ESP's payload data is simply a message for a higher layer such as UDP or TCP.
    - In this mode, IPsec acts as an intermediate protocol layer, much like SSL/TLS does between TCP and a higher layer.
    - When an ESP message is received, its payload is passed to the higher level protocol.
  - In a tunnel mode SA, however, ESP's payload data is itself an IP packet

Example Systems

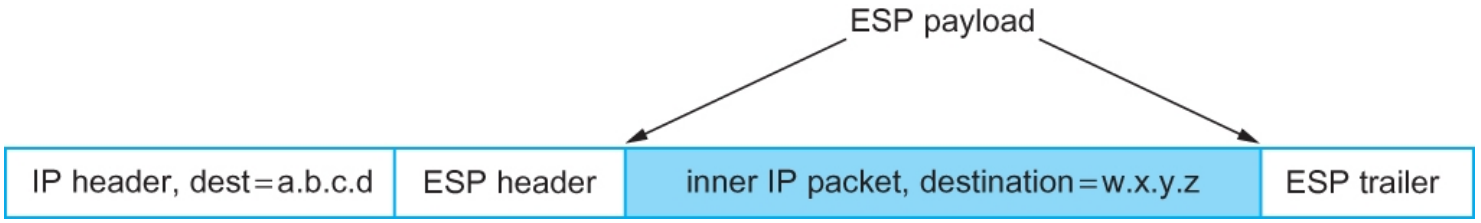
- IP Security (IPSec)



IPsec's ESP format

Example Systems

- IP Security (IPSec)



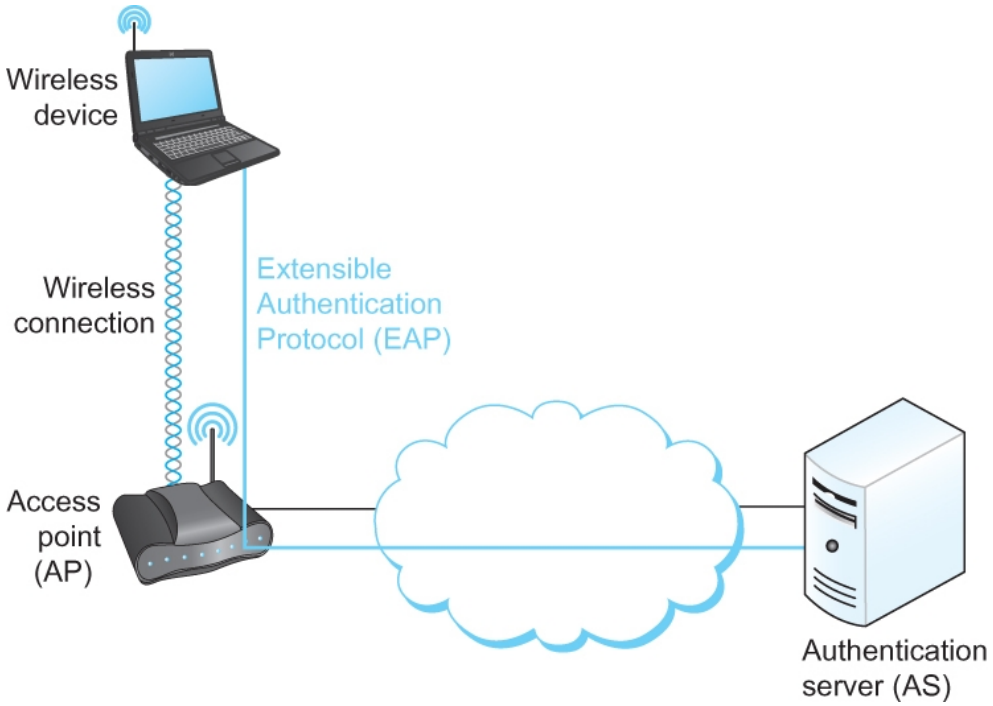
An IP packet with a nested IP packet encapsulated using ESP in tunnel mode. Note that the inner and outer packets have different addresses

Example Systems

- Wireless Security (IEEE 802.11i)
  - The IEEE 802.11i standard provides authentication, message integrity, and confidentiality to 802.11 (Wi-Fi) at the link layer.
  - *WPA2 (Wi-Fi Protected Access 2) is often* used as a synonym for 802.11i, although it is technically a trademark of The Wi-Fi Alliance that certifies product compliance with 802.11i.
  - 802.11i authentication supports two modes. In either mode, the end result of successful authentication is a shared Pairwise Master Key.
    - *Personal mode, also known* as *Pre-Shared Key (PSK) mode, provides weaker security but is more convenient and* economical for situations like a home 802.11 network.
    - The wireless device and the Access Point (AP) are preconfigured with a shared *passphrase—essentially a very long* password—from with the Pairwise Master Key is cryptographically derived.

Example Systems

- Wireless Security (IEEE 802.11i)



Use of an Authentication Server in 802.11i

Firewalls

- A firewall is a system that typically sits at some point of connectivity between a site it protects and the rest of the network.
- It is usually implemented as an “appliance” or part of a router, although a “personal firewall” may be implemented on an end user machine.
- Firewall-based security depends on the firewall being the only connectivity to the site from outside; there should be no way to bypass the firewall via other gateways, wireless connections, or dial-up connections.

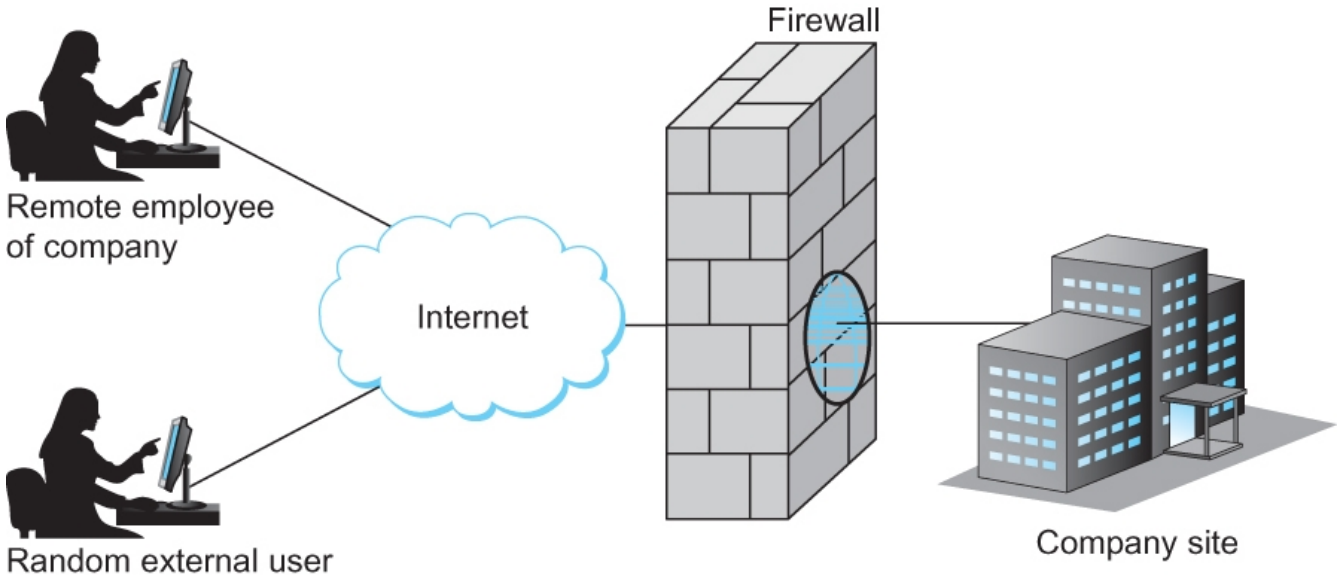
Firewalls

- In effect, a firewall divides a network into a more-trusted zone internal to the firewall, and a less-trusted zone external to the firewall.
- This is useful if you do not want external users to access a particular host or service within your site.
- Firewalls may be used to create multiple *zones of trust, such as a hierarchy of* increasingly trusted zones.
- A common arrangement involves three zones of trust: the internal network; the *DMZ ("demilitarized zone");* and the rest of the Internet.

Firewalls

- Firewalls filter based on IP, TCP, and UDP information, among other things.
- They are configured with a table of addresses that characterize the packets they will, and will not, forward.
- By addresses, we mean more than just the destination's IP address, although that is one possibility.
- Generally, each entry in the table is a 4-tuple: It gives the IP address and TCP (or UDP) port number for both the source and destination.

Firewalls



A firewall filters packets flowing between a site and the rest of the Internet

Summary

- We have discussed privacy and security issues in the network
- We have discussed different authentication protocols
- We have discussed different key distribution protocols
- We have discussed different cipher techniques
  - Classical and Public-Key
- We have discussed some examples of secured systems
  - PGP, SSH, IPSec