Securing TCP/IP

Chapter 11

Cryptography

- **Means "secret writing". Scramble the message so it cannot be read**
- **Several uses of cryptography**
  - **Hashing**
  - **Shared Key Cryptography**
  - **Public Key Cryptography**
  - **Digital Signatures**
  - **Digital Certificates**

Hashing

- **One-way. Once plaintext has been hashed, it cannot be recovered**
- **No key**
- **Used for passwords and to create a message digest**
- **Creates a fixed-length hash value from a variable length plaintext**

Password Hashes

- Passwords are stored as hashes so that if an attacker steals the password file, he does not get all of the passwords
- When the user enters his password, it is hashed and compared to the stored value

Hashing for Integrity

- **Hash the data being sent – creates a hash (sometimes called a digest)**
- **Digest is sent along with the original data, which is usually in plaintext**
- **Hashing is used only for integrity to ensure that:**
  - **Information is in its original form**
  - **No unauthorized person or malicious software has altered the data**
  - **Does not provide confidentiality**

Hashing Defeats Man-in-the-Middle Attacks

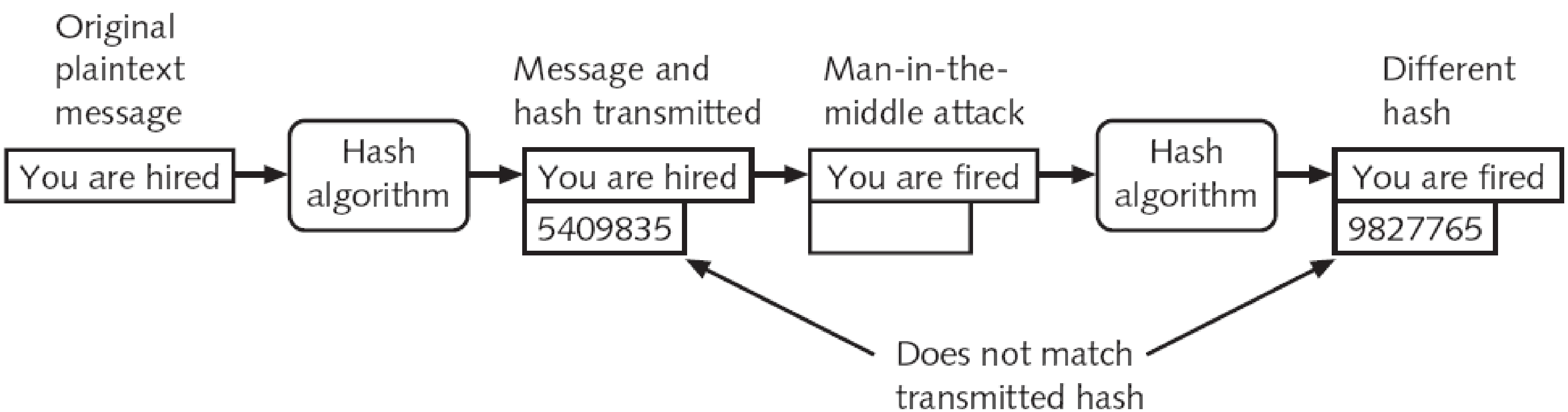


Figure 11-4 Man-in-the-middle attack defeated by hashing

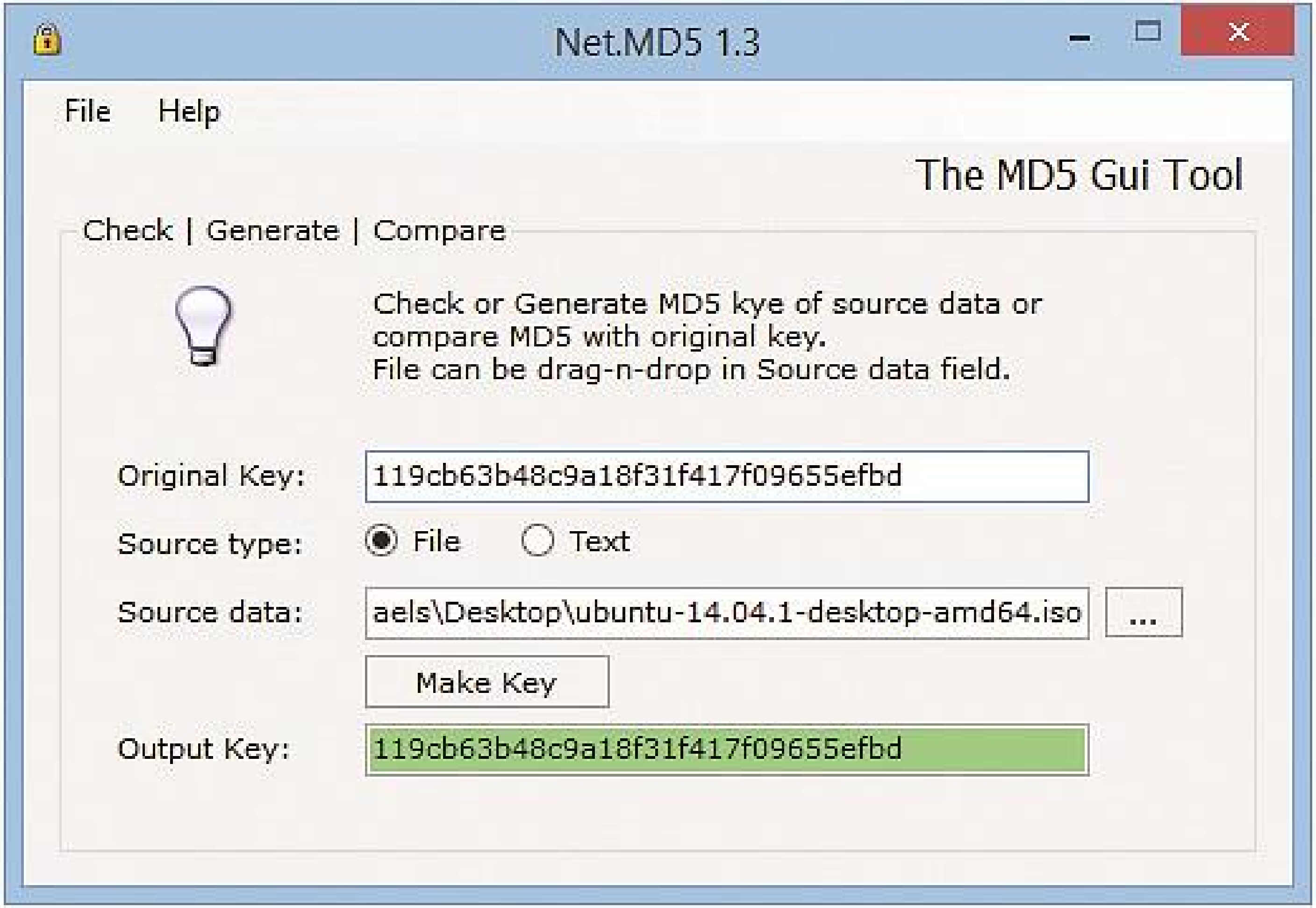


Figure 11.11 File and MD5

Caesar's Cipher

- Used by Julius Caesar over 2000 years ago, but still a valid cipher
- Replace each letter with the letter x letters after it in the alphabet
- Example: Replace a with c, b with d, c with e, d with f, etc

Caesar's Cipher

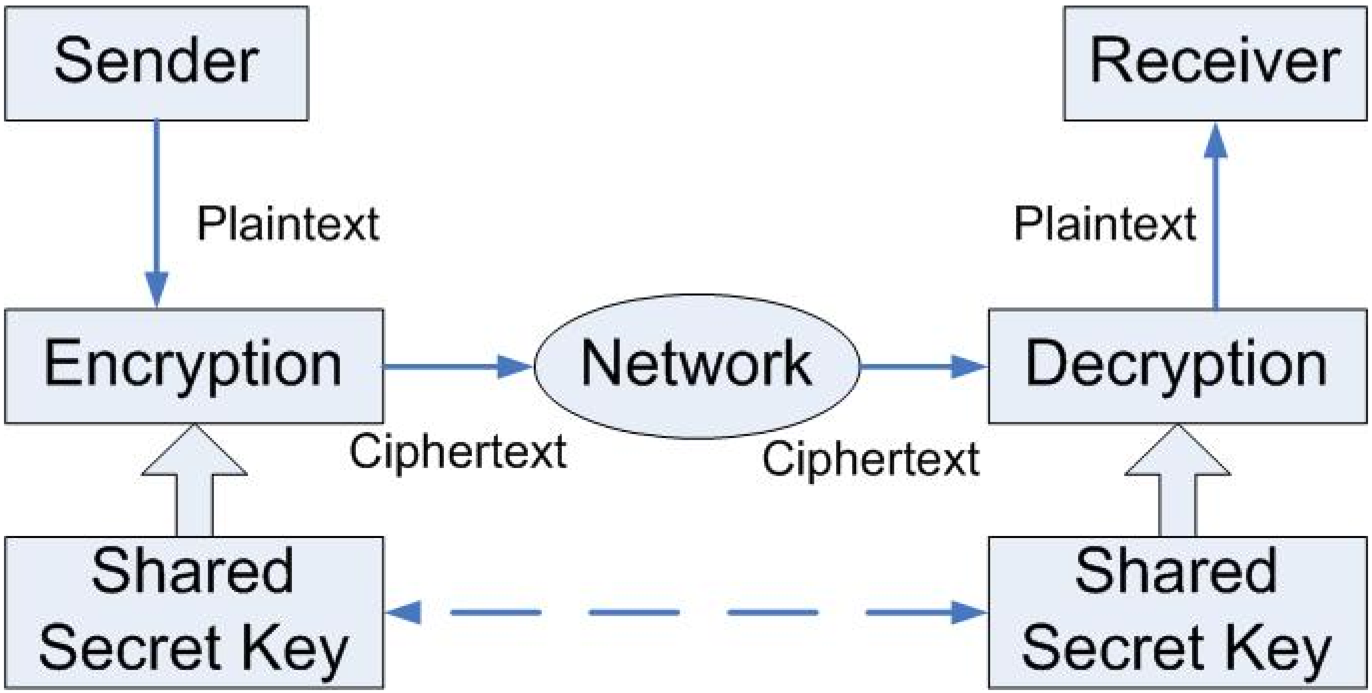
```
a b c d e f g h i j k l m n o p q r s t u v w x y z
r s t u v w x y z a b c d e f g h i j k l m n o p q
```

- To encrypt "monkey", replace each letter in the first line with the corresponding letter in the second line – it becomes "dfebvp"
- To decrypt "dfebvp", replace each letter in the second line with the corresponding letter in the first line – it becomes "monkey"

Symmetric Key Cryptography

- **Caesar's cipher is an example of symmetric key cryptography. (Also called shared key cryptography)**
- **The encryption and decryption algorithm (replacing one letter with another) is public but the key (which specific letter replaces another) is private**
- **The same key is shared by both parties**
- **Difficult to distribute the key to both parties while still keeping it secure**

Symmetric Key Cryptography



Keys

- Caesar's cipher is easy to crack because there are only 25 possible keys.
- In more modern ciphers, the key is a number that is used by the encryption and decryption algorithm.
- The more digits in the key, the more secure the encryption. (Because there are more possible values of the key, making it harder to guess.)

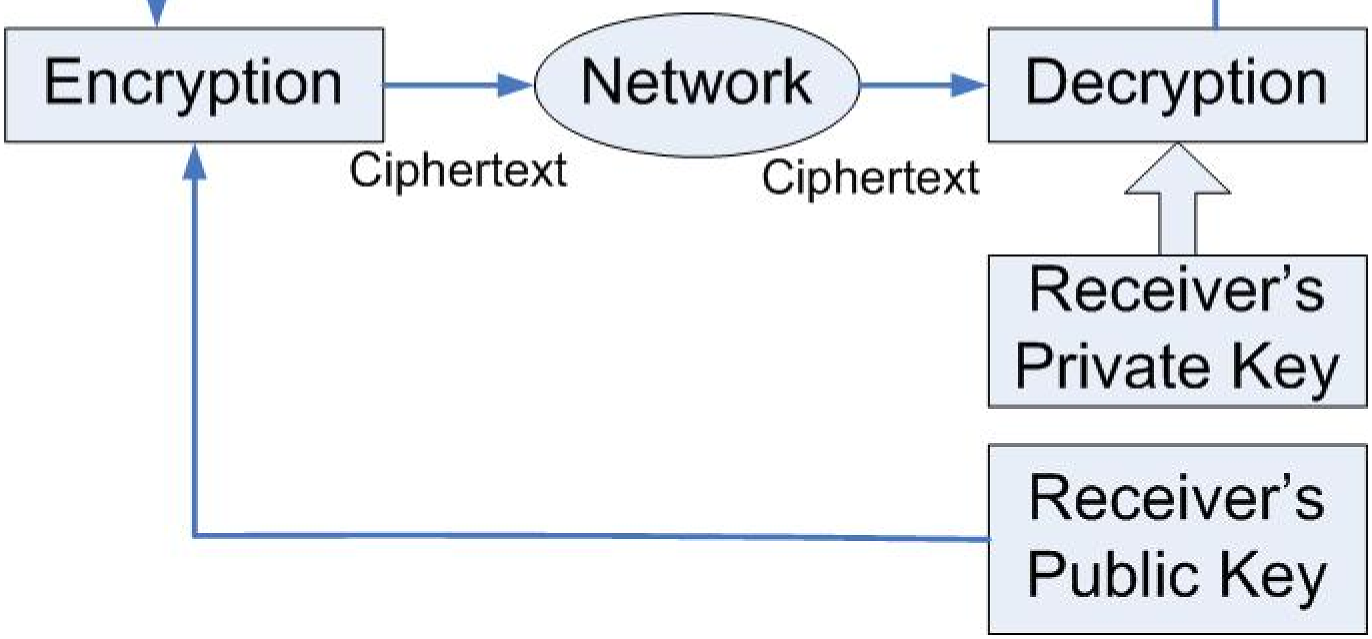
Public Key Cryptography

- **All ciphers used shared key cryptography up until about 40 years ago.**
- **Public key cryptography uses two keys – a public key and a private key**
- **Public key cryptography is a significant advance because the parties do not need to agree on a key in advance**

Public Key Cryptography

- Two keys - public key and private key are different, but mathematically related
- Receiver makes his public key available on a web site or in his e-mail messages
- Sender uses the public key to encrypt a message
- Receiver then uses private key to decrypt
- Only receiver ever sees his private key
- Just as in shared-key encryption, the encryption and decryption algorithms are public

Public Key Cryptography



Public Key Cryptography

- Based on factoring large numbers into prime factors
- Public key is product of two large prime numbers, p * q (p and q are very difficult to calculate from the product)
- Public key used to encrypt and private key, which knows p and q, is used to decrypt

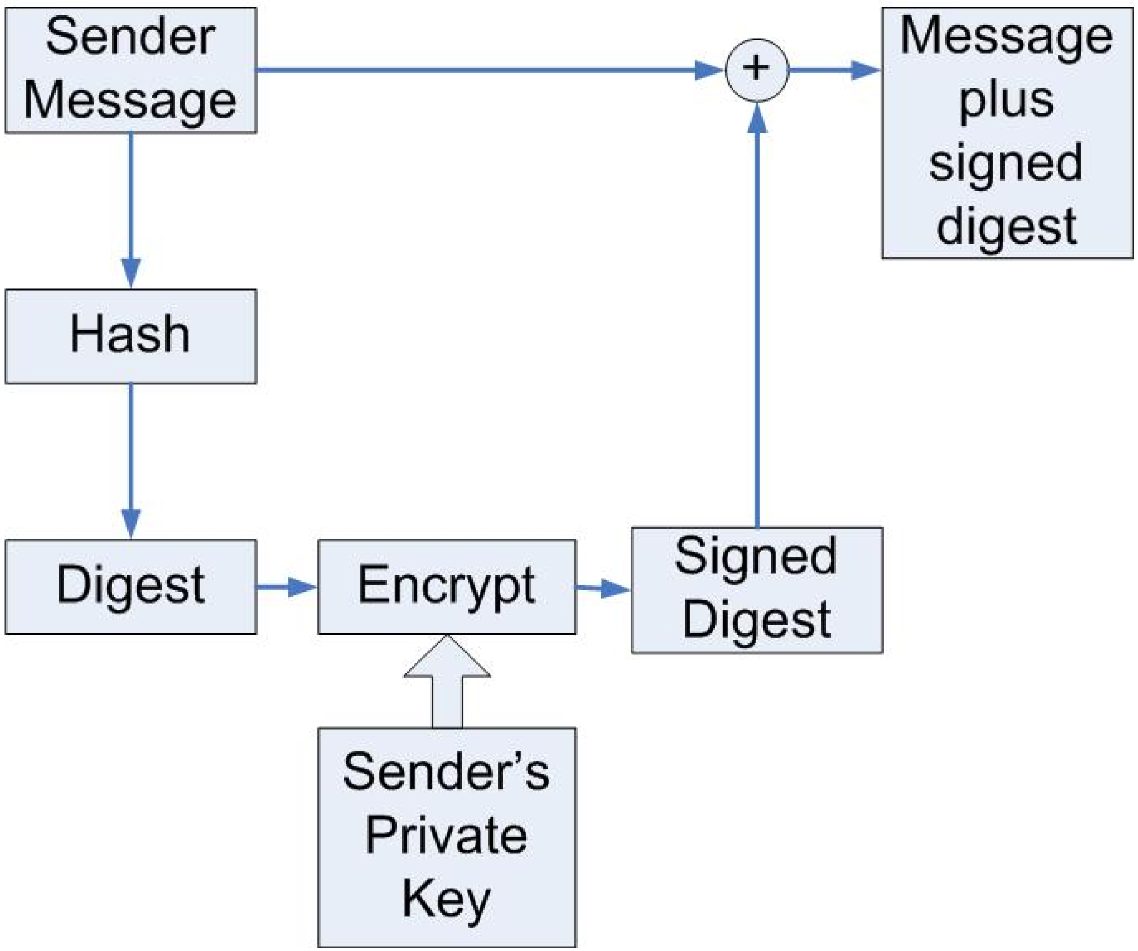
Public Key Cryptography

- **Anything encrypted with the public key can be decrypted using the private key**
- **Anything encrypted with the private key can be decrypted using the public key**

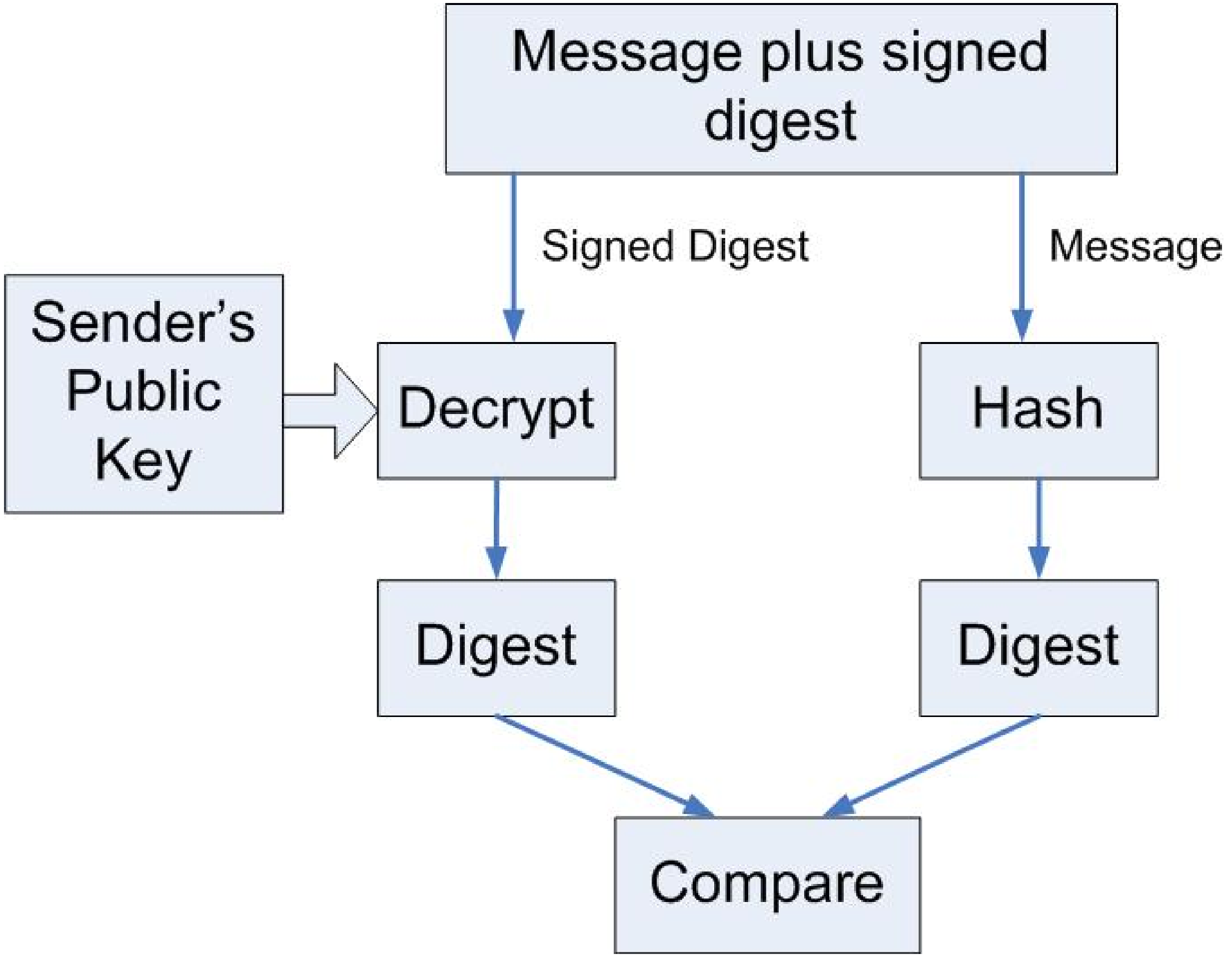
Digital Signatures

- Alice receives a message from Bob that was encrypted with her public key
  - She cannot know that the message was actually from Bob because anyone can use her public key
- Solution: Digital Signatures
- A digital signature can:
  - Verify the sender
  - Prove the integrity of the message
  - Prevent the sender from disowning the message

Digital Signature at Sender



Digital Signature at Receiver



Digital Certificates

- There is still a weakness with digital signatures
- To use a digital signature to determine that Bob is the sender, Alice must retrieve Bob's public key
- How can she be sure that it is Bob's public key and not one posted by an imposter under Bob's name?
- Digital Certificates!

Digital Certificates

- Associate or bind a user's identity to a public key
- A Digital Certificate is issued by a reputable third party, called a Certificate Authority
- Certificate basically says "I certify that public key a9856b…… belongs to Bob Smith"
- The digital certificate is digitally signed by the Certification Authority using its private key on a hashed digest of the certificate

Digital Certificate

I hereby certify that the public key
    19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
    Robert John Smith
    12345 University Avenue
    Berkeley, CA 94702
    Birthday: July 4, 1958
    Email: bob@superdupernet.com

SHA-1 hash of the above certificate signed with the CA's private key

Server Digital Certificate Example

- The web server administrator creates public and private keys and a digital certificate
- When user submits a credit card number through a web page, the server presents the certificate to the browser
- Browser checks hashed digest and verifies that it recognizes the CA
- Web server's public key (in the certificate) used by the browser to encrypt the credit card number
- Web server decrypts credit card number with its private key

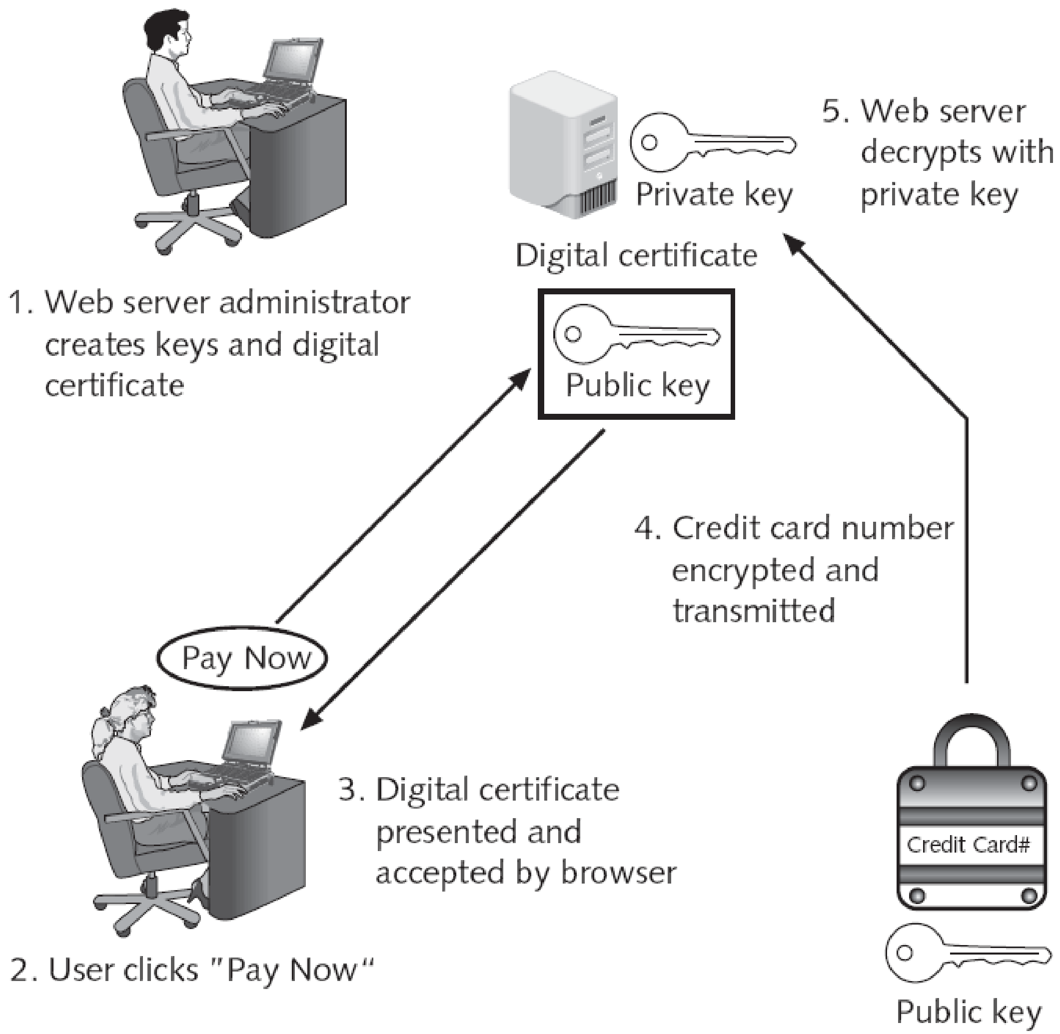


Figure 12-4 Server digital certificate

Authorization

- Assign levels of access to resources
- Note that a legitimate user may not have authorization to use all resources
  - You can't see my files, for example
- Access control list (ACL)
  - List of permissions
  - What an authenticated user may do

Authentication

- **Verify legitimate right of access to the network, based on what a user HAS, what a user KNOWS, WHO a user is, or WHERE a user is**
  - **Use cards, keys, and badges (what a user has)**
  - **Use PINs and passwords (what a user knows)**
  - **Use physical traits, such as fingerprints or retinal scans, for identification (who a user is)**
  - **Use geolocation to make sure user is where he is expected to be (where a user is)**

Authentication Servers

- Consider the situation where we have many users trying to log on to many different servers on a network using a username and password
- One solution is for every server to know every user's password
  - Inefficient - to change his password, user must log in to every server
  - Insecure - if someone breaks in to one server, all users are compromised

Authentication Servers

- **Better solution is one trusted authentication server that can grant access to any server**
  - Convenient
  - More secure
  - Single point of failure
  - Also usually provides encryption key management
  - Expense of server, plus support time
- **Used in larger networks**

Authentication Servers

- The most common type of authentication servers are
  - RADIUS (Remote Access Dial In User Service)
  - Kerberos

RADIUS Server

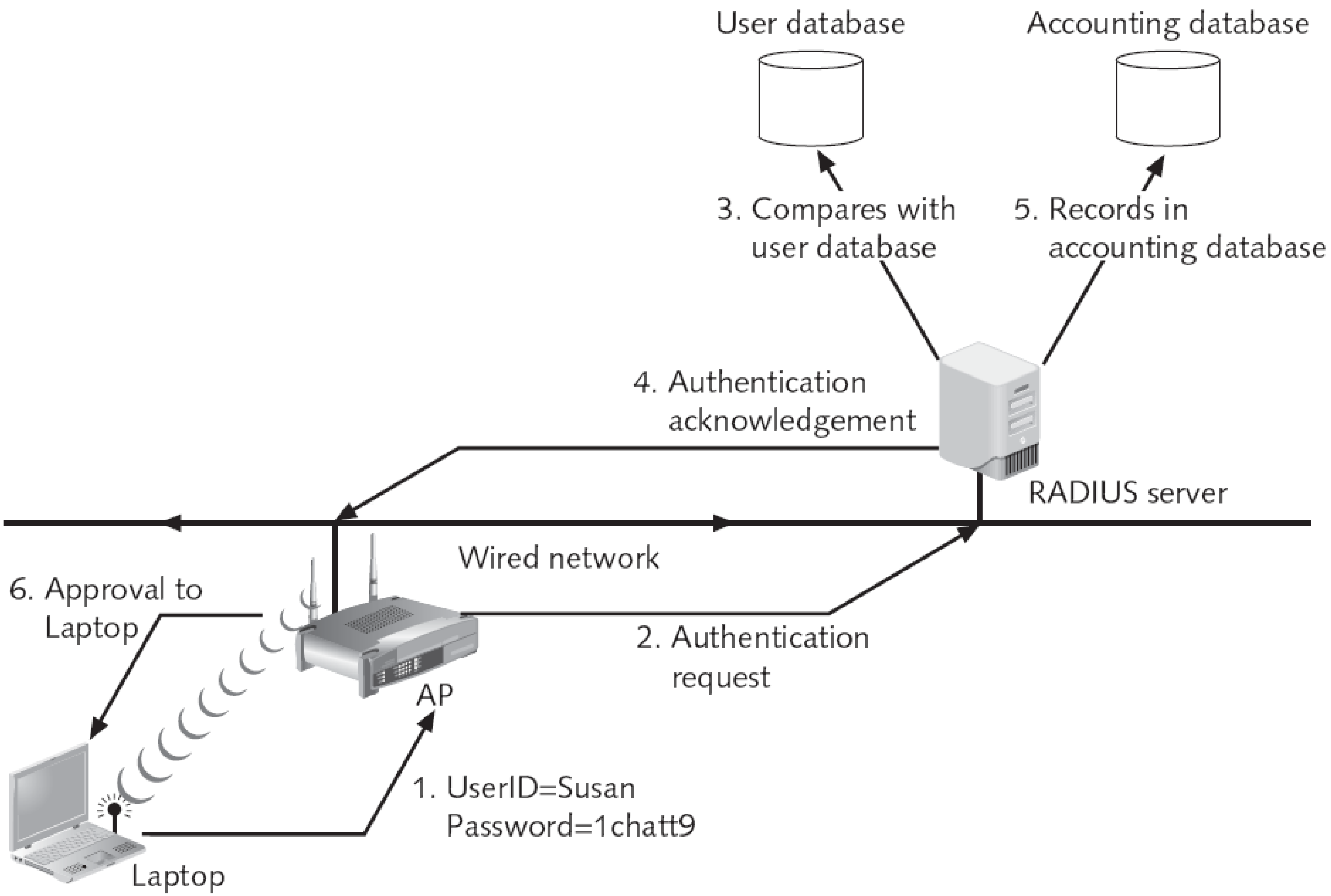


Figure 8-8 RADIUS authentication

Encryption and the OSI model

- **Encryption at different layers of the OSI model**
  - Layer 1: No common encryption at this layer
  - Layer 2: Proprietary encryption devices
  - Layer 3: IP Security (IPsec) protocol
  - Layer 4: No encryption methods for TCP or UDP
  - Layer 5: Important encryption standards (e.g., SSL and TLS used in e-commerce) happen within these layers, but don't fit cleanly into the OSI model

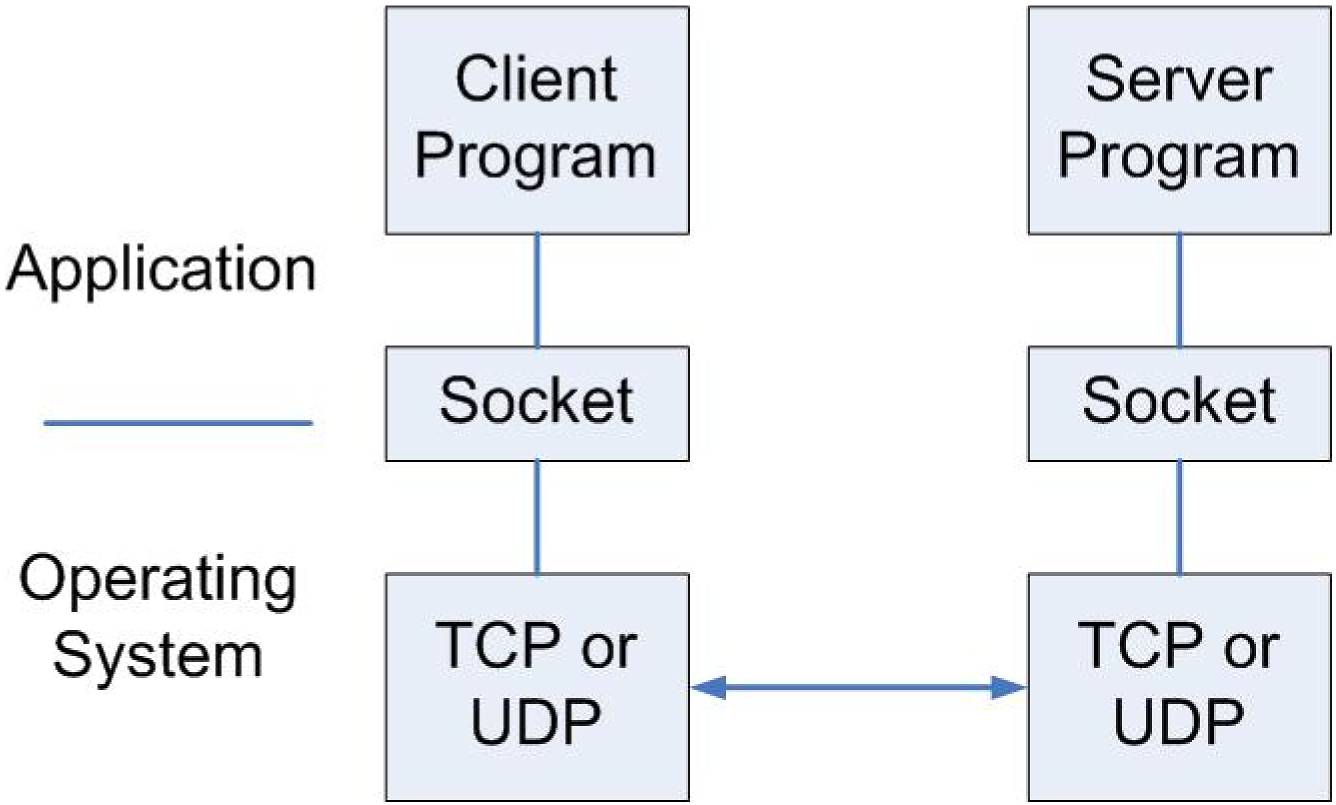
Application Layer Security

- Some argue that security should be provided at the application layer.
- This means that all intermediate devices and lower-level protocols automatically checked
- The problem is that each application protocol (http, smtp, pop3, ftp, etc) would have to be changed to add security
- Another problem is that each application is different, so security would be different on each

Application Layer Security

- An alternative way to provide application layer security is to build security into sockets
- A socket is the interface between the application layer and the transport layer in the host
- Sockets are an API, so they look like calls in the application program
- All the advantages of application layer security, but only need to change the sockets

Sockets


Application
Operating
System
Client
Program
Socket
TCP or
UDP
Server
Program
Socket
TCP or
UDP

Secure Socket Layer

- SSL (secure socket layer)
- Originally designed by Netscape
- Two main functions
  - Handshake protocol to establish a secure connection
  - Data exchange protocol
- TLS (Transport Layer Security) is an extension of SSL that is intended to supersede it
- HTTPS (Secure HTTP) uses SSL/TLS

Network Layer Security

- Some argue that security should be provided automatically by the network layer so that the user or application programs need not be involved
- Each individual IP packet is encrypted
- Network layer provides security using IPSec

IP Sec

- Set of protocols that provide authentication and privacy services at the IP layer
- IPSec developed for IPv6, but was also backfit into IPv4.
- Flexible
  - does not restrict authentication or encryption algorithms
- Since this must run quickly, uses symmetric key cryptography

IPSec

- Provides three security features
  - Authentication. Authentication header used to verify that packet was sent from the source address in the IP header and that it has not been changed.
  - Confidentiality. Encryption and the Encapsulating Security Payload header used.
  - Key management. Internet Security Association and Key Management Protocol (ISAKMP) handles the shared keys. Complex.

Secure Application Layer Protocols

- SSH – a secure replacement for Telnet
- HTTPS – a secure replacement for HTTP
  - Uses SSL/TLS for authentication and encryption
- SFTP – A secure replacement for FTP