Security on the Internet

Lecture Outline

- Security Risks on the Internet
- Cryptography
- Digital Signatures
- Message Digests
- Digital Envelopes
- Digital Certificates

Introduction

- Security for systems on the Internet is a very important area, since the Internet involves putting confidential data onto a space which can potentially be accessed by anyone.
- It also covers a very wide area, most of which requires very in depth study.
- In this unit, we will be going through some of the preliminary concepts of security in relations to some the web technologies you have seen so far.

Introduction (cont'd)

- When looking at security on the Internet, we consider the risks from three points of view:

1. The client (or user) side
   - active content, privacy infringement.
2. The server side
   - webjacking, LAN break-ins, denial-of-service.
3. Both the client and the server
   - eavesdropping, fraud.

Risk: Eavesdropping

- Eavesdropping is the concept of an unauthorized third-party listening to communication between a client and a server.
- This is a high risk since any communication between a client and server on the Internet typically goes through many different networked machines and systems.
- Neither clients nor servers can control the security of the intermediate systems.

Risk: Eavesdropping (cont'd)

- If a *packet sniffers* software can break into any single system in the connection, and will be able to listen to all communication between the client and server.
- More dangerously, the data being transmitted could also be altered in minor ways undetectably.
- *Encrypting* the data being transmitted is the main defence against eavesdroppers.

Risk: Fraud

- Fraud is the danger of someone pretending to be someone else.
- Examples:
  - A unauthorized user trying to access a private server
  - A web site pretending to be a teller system for a bank and getting account information from users
- When exchanging confidential information between users and server, both parties have to be confident that they are communicating with the right parties.

Risk: Fraud (cont'd)

- To combat fraud, there needs to be *authentication* between the clients and servers before exchanging information.
  - Eg. using "digital signatures" (more on this later).

Risk: Active Content

- These days, with uncertainty as to whether content of downloaded documents are *active* or *passive*, the danger of malicious active content is very big.
- *Passive content* are basic data that do not do anything once downloaded.
  - Eg. text.
- *Active content* are data that when downloaded within a certain environment (eg. Java-enabled web browser) will start some processing on the client machine
  - Egs applets, scripts, animations, data for plug-ins, etc.

Risk: Active Content (cont'd)

- Some apparently passive content can turn into active when used in certain way.
  - Eg. WORD documents with macro viruses is only active when opened in MS-WORD with macros enabled.

Risk: Privacy Infringement

- All web communications (HTTP, cookies) involves users giving the server certain information about the user
  - client browsers, what the user accessed, time of access, etc.
- Users cannot control how the server uses this information.
- More serious infringement when user voluntarily gives away extra information for particular purposes (online surveys, email messages, online order forms) but the information is used in unauthorized ways.

Risk: Privacy Infringement (cont'd)

- The need for privacy in a lot of cases conflicts with the need for very strong authentication.
  - For authentication, we want the client to supply as much information as possible.
  - For privacy, we want the client to supply as little information as possible.

Risk: Webjacking

- The danger of someone breaking into a web site and vandalising the web site.
  - Deleting critical contents
  - Putting up malicious and crude messages
  - Opening access to confidential information
- Can leave much longer lasting damage than the vandalised content.
  - Eg. reputation for organisations where security is critical, like banks and government offices.

Risk: LAN Break-ins

- Danger of unauthorised access to other machines on the local LAN.
  - Most corporations and large organisations have important data stored on various parts of their local LAN.
  - Access to one of the machines on the LAN (eg. the file and web server) means access to the rest of the LAN.
- Protection usually in the form of *firewalls*.
  - Filters inappropriate packets coming from outside the LAN.
  - Block dubious packets coming from inside the LAN.

Risk: Denial-of-Service

- Web and Internet services can be made unusable if the server or the operating system is made to
  - crash,
  - hang, or
  - process requests so slowly
- Commonly done by making servers process large blocks of data, or process a large amount of requests.

Cryptography

- Cryptography is the science of transforming data into a form which is not easily readable by unauthorized parties.
- All cryptographic systems has 4 parts:
  - *Plaintext*: the data before encryption
  - *Ciphertext*: the data after encryption
  - *Cryptographic algorithm*: the operations of converting plaintext to ciphertext, and vice-versa
  - *Key*: a secret text string used to encrypt the plaintext, which someone MUST have to be able to decrypt the ciphertext.

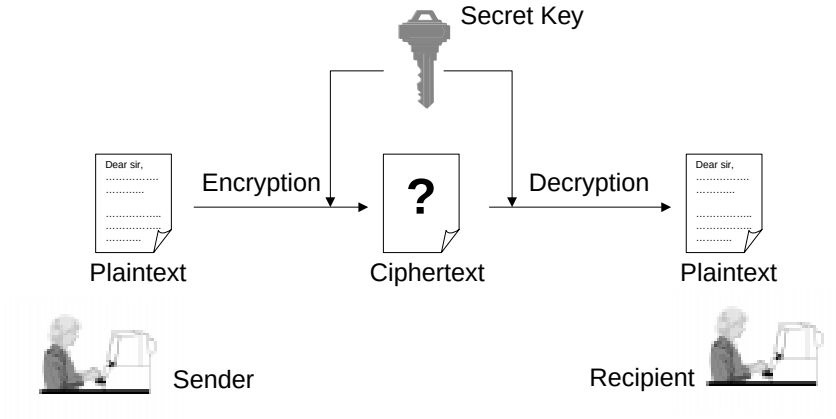
Encrypted Data

- Ciphertext can be transmitted over insecure and public lines.
- For good encryption algorithms, algorithm can also be made public
  - only the key needs to be kept secret.

Symmetric Cryptography

- The basic encryption system is symmetric, in the sense that the same key is used to encrypt and to decryption.
- Some common ones are:
  - DES (Data Encryption Standard)
  - RC2, RC4, RC5 – invented by RSA Data Security Inc.
  - IDEA (International Encryption Algorithm)

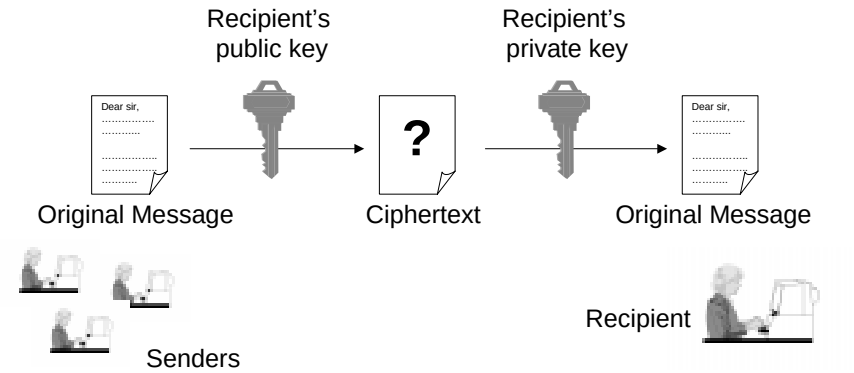
Symmetric Cryptography



Public Key Cryptography

- In public key cryptography, keys come in pairs
  - one for encryption, another for decryption.
- The encryption key is made public, but only the recipient has the private decryption key.
- Anyone wanting to send a message to the recipient will:
  1. Look up the recipient's public encryption key
  2. Encrypt with the public key
  3. Send the message to the recipient to decrypt using the private key

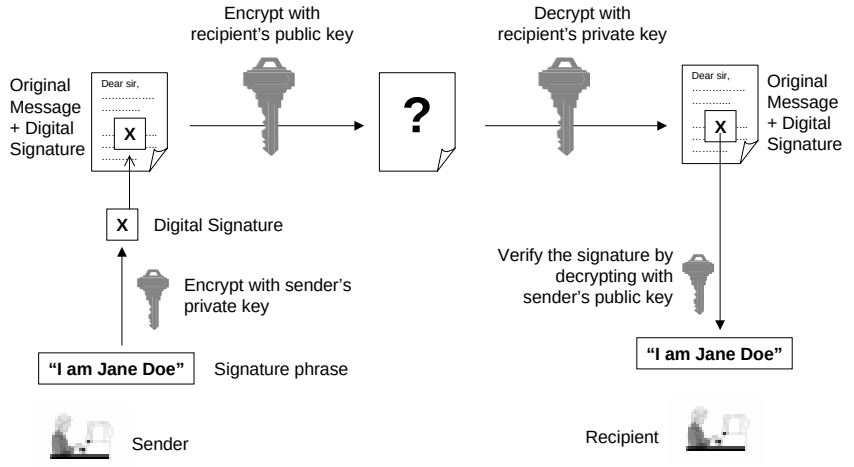
Public Key Cryptography



Digital Signatures

- Using public key cryptography, we can create unforgeable *digital signatures* for authentication.
- In digital signatures, another set of steps involving messages encrypted with a private key and decrypted with a public key is added

Digital Signatures



Digital Signatures (cont'd)

- Sender only needs to know recipient's public key, and recipient only needs to know sender's public key.
- Sender can be confident ONLY the recipient can decrypt the whole message.
- Recipient can be confident ONLY the sender could have generate the digital signature.
- In real life, the recipient usually sends periodic "challenge" phrases to the sender to use as signature, so the signature is not the same every time.

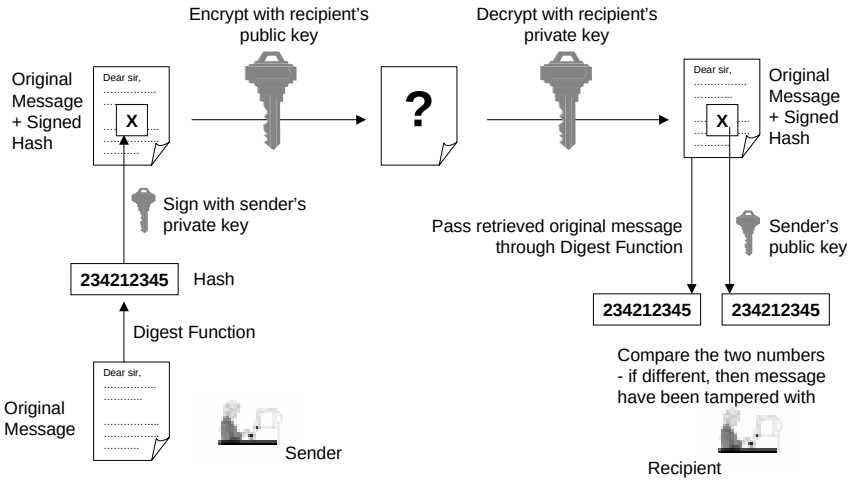
Message Digests

- Cryptographic methods should have *integrity checks*
  - to determine if the messages have been tampered with.
- The methods mentioned so far do not provide good integrity checks when messages are encoded in small text blocks
  - some blocks could be deleted or duplicated and the whole message will still decrypt.

Message Digests (cont'd)

- Message digest functions creates a "hash" number based on the message, but even a minor change in the message will give a very different hash.
- The digest (or hash) function works one way
  - you cannot get back the original message from the hash number.

Using Message Digest



Common Digest Functions

- MD4
  - produces 128-bit hashes (ie. binary hash numbers which are 128-bits long)
- MD5
  - also produces 128-bit hashes.
  - Created to solve some weaknesses found in MD4
  - The most widely used digest function today.
- SHA (Secure Hash Algorithm)
  - Produces 160-bit hashes
  - Designed by the U.S. National Institute of Standards and Technology (NIST) for its Digital Signature Standard (DSS)

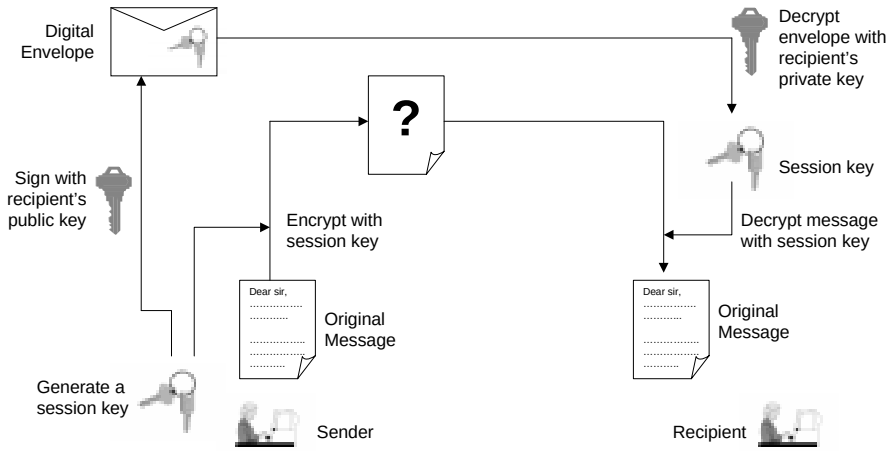
Digital Envelopes

- Public key cryptography is more suitable for the Internet than symmetric cryptography
  - no prior arrangement necessary for shared keys between sender and receiver.
- But public key systems are orders of magnitude slower in processing than symmetric systems
- Solution: combine the two systems using *digital envelopes*.

Using Digital Envelopes

- Steps:
  1. Sender generates a secret key (called the *session key*) at random
  2. Use the key with a symmetric algorithm to encrypt a message
  3. Encrypt the session key with the receiver's public key, to get the *digital envelope*
  4. Send the encrypted message and the digital envelope to the recipient
  5. Receiver decrypts the digital envelope using the receiver's private key, to get back the session key
  6. Receiver uses the session key to decrypt the message (with the same symmetric algorithm)
  7. Communication in the rest of the session between this sender and receiver uses the same session key
  8. When the session is finished, the session key is discarded.

Using Digital Envelopes



Certifying Authorities and Digital Certificates

- Another problem with public key cryptography: we need to know the public key of everyone we want to send secure messages to.
- Solution: to have trusted third parties acting as *certifying authorities* (CA)
  - senders/recipients only need to know CA's public keys.
- Concept: Before sending a message to the recipient, the sender requests a *digital certificate* from the recipient to verify they are who they claim they are.
  - These digital certificates must be signed by one of the CAs.

Getting Digital Certificates

- Steps to get a certificate:
  1. A party generates public/private key pair
  2. Sends public key and some identifying info to the CA, with payment to the CA for its services - a *certificate request*.
  3. CA checks and verifies receiver
  4. CA creates a new certificate containing the receiver's public key, the receiver's identifying info, a new hash and signs the certificate with the CA's own private key
  5. CA sends the requester the certificate and hash to complete its services

Using Digital Certificates

- Communication using digital certificates:
  1. Sender asks a receiver for a certificate
  2. Receiver sends the certificate to the sender
  3. Sender decrypts the certificate with the CA's public key
  4. Sender uses the CA's hash to determine if the certificate have been changed since the CA signed it – if not the sender uses the receiver's public key for subsequent communication

Different Digital Certificates

- *Site certificates*, for authenticating Web servers.
- *Personal certificates*, for authenticating individual users.
- *Software publisher certificate*, for authenticating genuine software; used by software companies to sign program executables.
- etc...

Example Secure Site Verification using Certificates

Go to URL https://ww.verisign.com

Click on the secure site image, and the Verisign authentication system will produce the site information for you to verify

Certifying Authorities

- Some recognized global CAs:
  - Verisign (http://www.verisign.com/)
  - Thawte (http://www.thawte.com/)
  - GlobalSign (http://www.globalsign.com/)
  - GTE CyberTrust (http://www.cybertrust.gte.com/)
- Some example Australian CAs:
  - Baltimore Certificates Australia (http://www.secdom.com.au/)
  - eSign Australia (http://www.esign.com.au/)

Public Key Infrastructure (PKI)

- PKI refers to the standards, products, services of certifying authorities, and the framework for the generation, distribution and management of public key certificates
  - Security companies (eg. Verisign) offers PKI as solutions for corporations.
- The Australian government has a PKI scheme called ***GateKeeper***, which defines standards for secure communication for the Australian government's online transactions.
  - See http://www.govonline.gov.au/projects/publickey/Gatekeeper.htm

References

- Most of the diagrams in this lecture comes from the book:
  - *Web Security: A Step-by-step Reference Guide,* Lincoln D. Stein, Addison-Wesley, 1998.
- Lincoln Stein is also the author of W3C's Security FAQ, more up-to-date than the book, available at:
  - http://www.w3.org/Security/Faq/
- For current security alerts, follow the links at:
  - http://www.w3.org/Security/Faq/wwwsf10.html

- Blank Page -