cis3.2
electronic commerce
24 april 2006
lecture # 22
internet security (part 2)

topics:

- finish from last time:
  symmetric (single key) and asymmetric (public key) methods
- different cryptographic systems
- electronic payment mechanisms
- security techniques
- firewalls
- secure sockets layer (SSL)

key distribution

- the most important weakness of symmetric encryption.
- How do computers A and B agree their keys?
- Options:
  1. A selects key and delivers it physically to B.
  2. A third party C selects the key and delivers it physically to both A and B.
  3. If A and B have previously used a key, they could send a new key using an encrypted message.
  4. If A and B each have an encrypted connection to a third-party C, C could deliver a new key to each of A and B on encrypted links.
- For link encryption, options 1 and 2 may be feasible. But these are not usually feasible for end-to-end encryption.
- Option 3 is vulnerable: If an opponent ever finds a key, all future communications can be read.

asymmetric (or public) key algorithms

- Similar to symmetric key encryption, but we use at least 2 keys: One for encryption (the public key), and one for decryption (the private key).
- The steps involved are:
  - Keys are generated in pairs, a public key and a private key, by each person or computer (say Bob).
  - The public key is made public (e.g., on a web-site) by Bob.
  - Anyone who wants to send a message to Bob, uses his public key.
  - Bob decodes the message using his private key.
- This approach dates from 1976.
- Public key methods are not necessarily more secure than symmetric algorithms. But there is a larger computational overhead, and this makes brute-force attacks harder to execute successfully.

requirements for public key algorithms

- It is computationally easy for party B to generate a pair of keys.
- It is computationally easy for sender A to generate the cipher text on the basis of the plain text and the public key.
- It is computationally easy for party B to decrypt the resulting ciphertext using his private key and so generate the plain text.
- It is computationally infeasible for an opponent to determine the private key from the public key.
- It is computationally infeasible for an opponent to recover the original plain text from the public key and the ciphertext.
- In addition, we may require (not necessary but nice to have):
  Either of the two related keys may be used for encryption with the other used for decryption.

applications of public key methods

- Encryption —
  sending coded messages.
- Authentication —
  when we want to be certain that the sender of a message is actually the person (or computer) they say they are. The sender of the message uses his private key to encrypt the message. Only his public key will be able to decode the message.
- Digital Signature —
  The sender "signs" a message using his private key. This application is similar to authentication.
- Key Exchange —
  Two parties co-operate to exchange a session key, using the private key of one or both parties.

public key algorithms

- RSA algorithm
  - Developed by Rivest, Shamir & Adleman at MIT in 1977
  - RSA is a block cipher in which the plaintext and cipher text are integers between 0 and (n-1) for some n. If M is the plaintext number, and C is the cipher text number, the algorithm works as follows:
    - Encryption algorithm: $C = M^e \ modulo \ n$
    - Decryption algorithm: $M = C^d \ modulo \ n$
  - Both sender and receiver must know the values of n and e. The public key is a pair of numbers (e, n).
  - Only the receiver knows the value of d. The private key is the pair of numbers (d, n).
- Digital Signature Standard (DSS)
  - A standard agreed in 1993 for digital signatures in the American National Institute of Standards.
  - Only used for digital signatures (not for encryption or key exchange).

authentication using asymmetric keys

- Here we use the public and private keys in reverse order to encryption.
- Alice sends a message to Bob
  - Alice encrypts the message using her private key, which only she knows.
  - Bob receives the cipher text, and decrypts it using Alice's public key.
  - If Alice really did send the message, the output should be plain text.
  - If someone else (say, Mary) sent the message, then Alice's public key will not work on this message.
- So, this provides a way to authenticate a message, assuming the private key has not been stolen or somehow made public.
- Note that this method does not keep the message secret. Anyone can use Alice's public key (since it is public!) to decode Alice's message.
- This approach is also used for Digital Signatures, analogous to personal signatures on checks.

how to distribute public keys?

- Answer is simple: put on your web-site, email your friends, shout it from the roof-tops!
- But if Alice gets an email from Bob telling her that 1023 is his public key, how does she know it really is his? Maybe someone is impersonating him and sending out a false key in his name!
- Digital Certificates seek to get around this. A user (e.g., Bob) presents his public key to a trusted third party and receives a digital certificate. The certificate contains a public key together with a a user ID for the key owner (Bob), all signed by the third party.
- Examples of third parties: Government agencies or a bank. The user (Bob) can then give the digital certificate to anyone else (e.g., Alice).
- A standard for digital certificates is X.509.

public key distribution of symmetric keys

- How do 2 parties share a symmetric key? These are also called secret keys, to distinguish them from public and private keys.
- They could deliver them physically (e.g., by courier).
- If they already share a secret key, they could send the new one by encrypted message.
- They could use public key certificates, as follows:
  1. Bob sends Alice his public key using a public key certificate.
  2. Alice prepares a message.
  3. Alice encrypts the message using one-off symmetric key for this session, e.g., she uses a session key.
  4. Alice encrypts the session key using Bob's public key.
  5. Alice attaches the encrypted session key to the message and sends it to Bob.

  Only Bob is able to decrypt the session key (since only he has his private key). So, only Bob can read the original message.

cryptographic systems (1)

Cryptography is used in several systems serving a variety of purposes

- Message Digest Functions
  - These produce a summary digest of a file, and can be used to see if the file has been altered.
  - Useful for detecting presence of viruses or tampering by opponents.
  - Sometimes used for message authentication codes (appended to a message, so that the receiver can see if the message was altered during transit).
  - Examples: HMAC, MD series (128 bit digest), SHA series (160 bit digest).
- Digital signatures
  - Unique identifier of a sender of a message
  - Can use public key cryptography in reverse
- Digital Certificates
  - Issued by trusted third party (e.g. bank, government agency) to verify user is who they say they are.

  - Usually third party's signature encrypted by the private key of the authorization party
    - So the receiver (Bob) needs to decode the authorization signature using the public key of the third party.
    - Then, if this works, use the public key of the sender (Alice) to decode her signature.

cryptographic systems (2)

- PGP
  - "Pretty Good Privacy"
  - A publicly-available system for encrypting files and email messages
  - PGP uses:
    - RSA for management of keys in symmetric encryption
    - IDEA algorithm for sending data using symmetric encryption
    - MD5 scheme for ensuring no tampering.
  - Main weakness: if a public key is compromised, than a revocation certificate has to be issued to everyone in contact with the person whose keys are compromised.
- PCT — Microsoft product similar to SSL (Secure Sockets Layer, to be discussed later)
- SHTTP — A secure version of HTTP. Did not take off.
- SET (Secure Electronic Transactions) — for credit card info (more later)

electronic payment systems

- Credit card schemes
  - Systems which enable customers to use their credit cards for payment of purchases over the Internet, and enable sellers to obtain authorization of these payments from credit card companies.
- E-wallets
  - A wallet holds a number of credit cards, so that the details do not need to be re-entered by the customer each time a transaction is to be made.
  - The e-wallet is usually stored encrypted on the customer's computer
  - e.g., SET is a standard for e-wallets
- Digital Cash or e-cash
  - An electronic version of cash or checks
  - A customer purchases credits from an e-bank or e-cash vendor, and then uses these credits to purchase goods from sellers. The sellers redeem the e-cash with the bank or e-cash vendor.

- Micropayment systems — These aggregate small Internet purchases and send a monthly bill to the customer

Secure Eletronic Transaction (SET)

- A standard for e-wallets
  - Developed by VISA, Mastercard, IBM, Microsoft, Netscape, RSA, and others.
  - Standard issued in 1997.
- Main advantages
  - Customer does not need to enter credit card details in for each transaction.
  - Details are kept on the client machine in an e-wallet.
  - Seller does not see credit card details of customer.
  - Credit card company does not see purchase transaction details (only the total amount).
  - All transmissions are secure

e-cash

- Cash payments have few of the risks of credit-card transactions.
  In particular, they are:
  - Anonymous: no record of who spent the money
  - Untraceable: no record of what it was spent on
  - Hard to counterfeit
  - Have limited maximum loss (no more than the cash value of coins/notes)
- Can we replicate these characteristics electronically?

electronic banknotes

- A basic form of electronic cash is easy to achieve via an *electronic banknote* digitally signed by the bank.
- Bank creates e-banknote, keeping its details secret, and digitally signs it with bank's secret key
- Bank issues e-banknote to client (Alice) and debits her bank account.
- Alice buys something from Bob, giving him the encrypted e-banknote.
- Bob contacts bank, which verifies the validity of the e-banknote and records that it has now been spent (and credits Bob's bank account).

properties of e-cash

This basic procedure provides some of what is needed:

- e-Banknotes are unforgeable (unless the bank's secret key becomes known)
- Same note cannot be spent twice (provided the bank keeps secure records of which ones have been spent).
- e-banknotes can be stolen
  - If signed note is copied before it is spent
  - But this can be avoided by using secure transactions.
- In the worst case, loss to the client (Alice) is limited to the value of e-banknote

privacy issues for e-cash

However, there is a privacy issue with this procedure:

- The bank is able to keep a record of which customers have been issued with an e-banknote and what they have spent them on.
- This can be prevented by issuing *blinded* e-banknotes.
- The principle is the same as for a blind signature on a check:
  - The bank signs an e-banknote in a way that cannot be forged, but without knowing the details of what it is signing.
  - This can be achieved with a public key algorithm.

B2B payments

- B2B payment mechanisms are similar to B2C mechanisms.
- In the 1980's, before the WWW, considerable attention was devoted to Electronic Data Interchange (EDI):
  - Standards for transmission of financial and accounting information between companies.
  - Most of these standards were too rigid for use in e-commerce transactions, and EDI is no longer a focus of great attention.
- One WWW system for B2B is Clareon
  - A buyer and seller agree to execute a purchase transaction.
  - The seller sends an invoice to the buyer over the Internet.
  - The buyer sends an authorization message to the Clareon system.
  - Clareon debits the buyer's account and credits the seller's account.
  - The Clareon system sends data to both buyer and seller. This is done in formats which are compatible with their respective computer systems.
  - All transmissions are encrypted.

some approaches to security

- Logging tools
  - software tools which monitor use of a computer
  - log particular events, e.g., user logins, transferring web-pages, attempts to access secure files
  - check for unusual events, e.g., access at unusual times, a user logging in and out repeatedly (could be seeking to gather info), a user mistyping a password (could be an attempted hack), a user accessing strange web-sites (e.g. military sites)
- Virus scanners — software which looks for unusual changes to files (especially operating system files)
- Security Checking software — used by system administrators to identify potential problems, e.g., scanners, password cracking software
- Network topology techniques
  - Design the topology of the network so as to make intrusion difficult
  - A common technique is a *firewall*

- This is an extra element placed between a network (or a network element) and the external world.

firewalls

- Functions of a firewall
  - To monitor traffic into (and sometimes out of) a private network
  - To reject traffic which is considered suspicious or unauthorized
- Components of a firewall
  - Router
    - To monitor traffic into the private network
    - To reject access from unauthorized users
    - To reject or reroute rejected packets, e.g., it may reject all emails with attachments, or just the attachments, or all those with undesirable content.
  - Bastion host (or proxy server)
    - To provide a temporary store (cache) of pages held on a real web-server
- Operation
  - When a request for (e.g.) a web-page is received by the router from some client, it it accepted or rejected.

- If accepted, it is passed to the proxy server.
  - If the page is in the cache, it is sent to the client who requested it.
  - If the page is not in the cache, the proxy server requests a copy from the real web-server.
    - When this is received by the proxy host, it is then sent to the client.

firealls: what protection do they offer?

- Any attacker can only reach the proxy server — e.g., deleting web-pages from the proxy just deletes them from the cache, not from the real web server.
- A stronger form of protection is a screened subnet. This has 2 layers of protection, with routers on each side.
- In the central zone, between the two routers, are proxy servers for various functions:
  - A proxy web-server
  - A proxy email server
  - etc.

  Each contacts a real server in the internal network

firewalls: types

- Packet-filtering router
  - Applies rules to each incoming packet and forwards or rejects them, one by one.
  - Filters may be on source or destination address fields.
- Application-level gateway
  - Clients attempt to use a specific TCP/IP application, such as Telnet or FTP.
  - To gain access past the firewall, they must enter access details (e.g., username and password).
  - If details are correct, the application is allowed to proceed; otherwise, not.
- Circuit-level gateway
  - This firewall does not permit an end-to-end TCP connection.
  - The firewall sets up two TCP connections:
    - One between it and the external client.
    - One between it and the internal client.
  - Messages are relayed across from one connection to the other.

  - Normally, message are not examined – the security occurs in setting up the connections.
- Typically, a bastion host is the platform for an application-level or a circuit-level gateway.

firewalls: limitations

Firewalls cannot protect against everything!!

- e.g., disgruntled employees on the inside network
- e.g., attacks which bypass the firewall
  - e.g., internal systems may have a dial-out facility to connect to an ISP.
  - e.g., a modem to allow travelling employees to dial-in to the internal network remotely.
- Firewalls cannot screen every type of email or request
  - Viruses and malicious code may still get past a firewall, particularly if they are not presented in the usual form (e.g. email attachments).

Secure Sockets Layer (SSL)

- SSL was developed by Netscape for Netscape Navigator (its browser)
- It operates at the levels between: (1) HTTP and FTP and (2) TCP/IP
- Main functions of SSL:
  - SSL server authentication
    - Enables a client to confirm the identity of a server.
    - Uses public key cryptography to validate the digital certificate of a server and confirm that it has been issued by a valid certification authority.
  - SSL client authentication
    - Enables a server to confirm the identity of a client
  - SSL encryption
    - Uses symmetric encryption to send data to/from servers/clients.
- There are 2 sub-protocols
  - SSL Record Protocol — used for transmission of bulk data
  - SSL Handshake Protocol — used to establish the keys and algorithms to be used for data transfer

The SSL process

- Phase 1: Handshake
  - To authenticate server
  - To authenticate client (optional)
  - To agree secret keys and algorithms for part 2.
- Phase 2: Data transfer.
- SSL uses public key cryptography for the handshake, i.e.,
  - To authenticate client and server
  - To establish keys and algorithms for encryption of data transfer.
- SSL uses symmetric key cryptography for encryption and decryption of data in the data transfer.