15.561A: Information Systems: From Technology Infrastructure to the Networked Corporation

15.566: Information Technology as an Integrating Force in Manufacturing

Class #15: TECHNOLOGIES FOR ELECTRONIC COMMERCE: SECURITY, ENCRYPTION AND PRIVACY

Spring 1998
Sloan School of Management
Massachusetts Institute of Technology

Yannis Bakos
E53-329
Tel. (617) 253-7097; Fax (617) 258-7579
Email: bakos@mit.edu
Web: http://web.mit.edu/bakos

SECURITY

2

What is it?

- **Secrecy**
  - ensure that only authorized users have access to computer and data resources
- **Availability**
  - ensure the computer services remain available to users in the face of partial failures
- **Accuracy**
  - ensure that multi-user access and system crashes leave data in an accurate state

SECURITY BASICS

3

- Vulnerabilities
- Threats
- Countermeasures

VULNERABILITIES: WHAT PARTS CAN BE COMPROMISED?

4

- Physical
- Hardware and software failures
- Media
- Emanation
- Communications
- Human

THREATS: WHAT CAN BREACH SECURITY

- Natural physical disasters
- Unintentional human activity
- Intentional human activity
  - Foreign agents
  - Terrorists
  - Criminals
  - Corporate competitors
  - Crackers
- Insiders or outsiders?

COUNTERMEASURES

- Access controls
  - Protects information in computers
- Encryption
  - Protects communications and compromised data
- Emanation shielding, physical locks, etc.
  - Protects physical access to computers

ACCESS CONTROL TECHNIQUES

7

- Something you have
- Something you know
- Something you are

VIRUSES AND OTHER CRITTERS

8

- Programs that run on machines where they're not wanted
- Transmitted through I/O channels
- Disguise themselves
  - –How?
- Often don't act right away
  - –Why not?
- Why hasn't anyone written a definitive virus eliminator?

SPOOFS

9

- Pretending to be someone else
- Hard to login without someone's password
- But can send out communications with someone else's name on it
  - email
    - Dartmouth 1993: a message was sent saying midterm exam was cancelled
    - Message appeared to come from Professor!
  - world wide web
    - can spoof the entire Web!

WHY BOTHER WITH ENCRYPTION?

10

- Security of Telecommunications
  - Cyberspace is replacing face to face
  - Encrypted "secure channels" over insecure communication media
- Supply Chain Integration
- Electronic commerce
  - transactions: contracting, payment
  - delivery of information goods

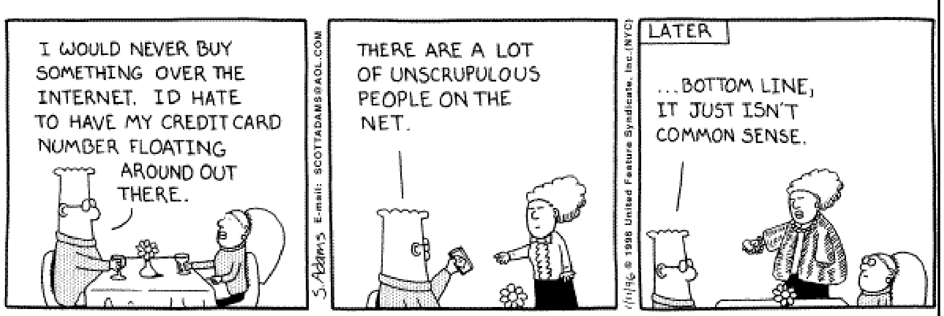
RISKS

11

- Abuse of information by dishonest people
  - –Theft
  - –Fraud
  - –Invasion of Privacy
  - –Cyber-Terrorism & vandalism
- Misuse by holders of private information
  - –Buying habits
  - –Medical history
  - –With whom you communicate

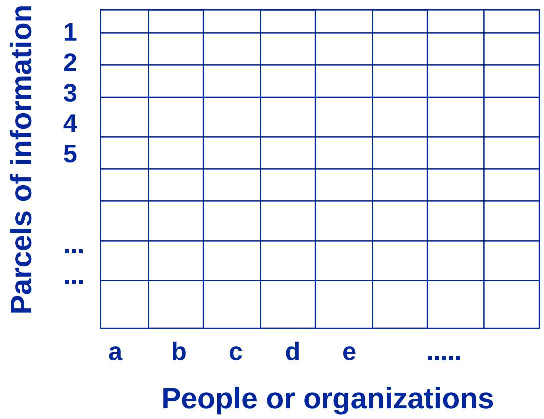
SECURITY: Internet vs. the Real World

12



13

THE POWER WE WOULD LIKE TO HAVE



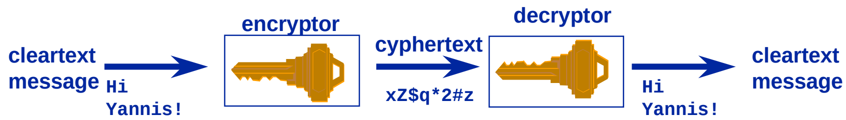
14

INSTRUMENTS AT YOUR DISPOSAL

- **Intellectual property law**
  - –Patents
  - –Copyright
  - –Trade Secret
- **Organizational innovations**
  - –isolation or encapsulation
  - –trust
- **Technology**
  - –dialback modems
  - –firewalls
  - –encryption

15

ONE SOLUTION: A CRYPTOSYSTEM



- Encryption and decryption machines typically use mathematical functions to convert between cleartext and cyphertext based on a "key"
- A good cryptosystem depends *only* on secrecy of key
- Two parties can use cryptosystem to establish a secure channel over an insecure network

16

ENCRYPTION AND DECRYPTION

- Alice and Bob would like to communicate privately
- Darth might be listening
- Alice represents (encrypts) her message as a ciphertext
- Bob interprets (decrypts) the ciphertext
  - converts it back into the original message ("plaintext")
- Darth can intercept the ciphertext, but doesn't know how to interpret it

SECRET KEY CRYPTOSYSTEMS

17

- First agree on a shared secret key, used for both encryption and decryption
  - Example: add "x" to each letter, where 'a'=1, 'b'=2, 'c'=3...
  - a key: x = 13
- System is good if:
  - All possible keys must be tried to read (or forge) messages - no "trap door"
  - Trying all keys takes "forever"
  - If a message decrypts properly, sender's identity is authenticated
- Good systems are:
  - hard to design
  - harder to verify

Transposition Ciphers

18

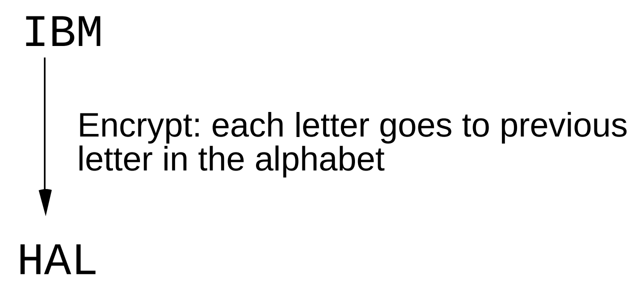
- Don't change any of the bits, just rearrange them

```
FOURSCORE AND SEVEN YEARS AGO
        |
        |   Get rid of spaces and
        |   arrange in three columns
        v
       FOU
       RSC
       ORE
       AND
       SEV
       ENY
       EAR
       SAG
       O
        |   Read down the columns
        |   instead of across
        v
FROASEESOOSRNENAAUCEDVYRG
```

Substitution Ciphers

19

- Substitute for each letter (block of bits)

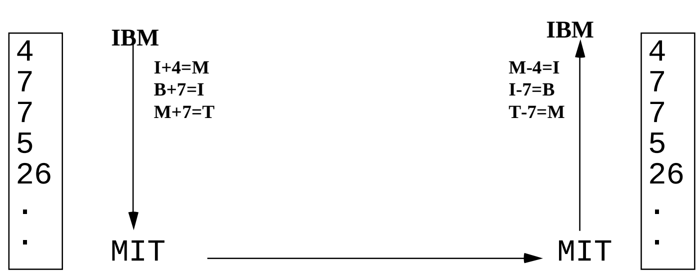


- How can you crack a substitution cipher?
  - i.e., how can you guess the key?

One Time Pads

20

- A substitution cipher, but the substitution method changes for each letter (block)
- Sender and receiver each get identical copies of a set of random numbers
  - Interpret number *n* as "substitute letter n later in alphabet"
- Provably unbreakable
- Problem is creating and distributing truly random one-time pads



EXAMPLE: SECRET KEY SYSTEMS IN USE

- **Data Encryption Standard (DES)**
  - **IBM & NSA in 1975**
    - widely used, not proprietary
  - **56 bit keys**
    - 256 = 7*1016 keys to pick from
    - Is this a large number?
  - **Triple DES: 112 bit key**
- **Skipjack**
  - **used in Clipper system (key escrow)**
  - **80 bit key**
- **Plus, lots of (mostly bad) proprietary systems**

More on the DES algorithm

- **DES = Data Encryption Standard**
  - **Developed by IBM in 1970s, with input from NSA**
  - **Official standard for non-classified government communications**
  - **De facto standard for financial transactions**
- **Private key system**
  - **Same key used for encryption and decryption**
  - **Key determines a sequence of permutations and substitutions**
  - **Process implemented in hardware; only keys are variables**
- **Some argue that NSA deliberately made DES weak**
  - **Keys are 56-bits long**
  - **IBM had another algorithm available that used 128-bit keys**
  - **But no one has publicly proven it's breakable**

Copyright © 1998 by Yannis Bakos
Spring 1998, Class #15

23

PROBLEM: KEY MANAGEMENT IS HARD

- Need to exchange secret key in advance
  - –same problem all over again
- Okay for:
  - –small scale communication
  - –own files
- Doesn't work as well for:
  - –secure interorganizational email
  - –encrypted phone/fax
  - –electronic commerce
  - –authentication with people you don't trust

MIT Sloan

24

PUBLIC KEY CRYPTOSYSTEMS

- Use a pair of keys: one encrypts, one decrypts
- Users publish one key, and keep other secret
  - –Look up recipient's public key, encrypt and send message
- Whole new ball game
  - –No prior arrangement needed
  - –If compromised, just publish new key!

MIT Sloan

Public Keys: Diffie-Hellman, RSA

25

- Each person has a pair of keys e for encryption and d for decryption
- Make e publicly available
- Alice uses Bob's e_B to send him a private message M^{e_B}
- Bob decrypts with d_B : $\left(M^{e_B}\right)^{d_B} = M$
  - No one else knows d_B
- Works as long as
  - d is really kept secret
  - Hard to compute d from e
  - Can get the correct e from some trusted source

EXAMPLE: PUBLIC KEY SYSTEMS IN USE

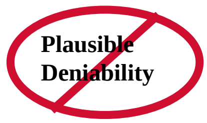
26

- RSA System
  - Rivest, Shamir & Adleman at MIT in 1978
  - Based on factoring really large numbers
  - Very slow
    - runs easily on PC cards
    - usually used in combination with secret key
    - example: RSA (for key) + DES (for message)
  - Challenge based on 129 bit key broken last year
    - How? 5-6 months with internetworked computers
    - Counter: add 3 bits to key & double factoring time!
- Other Systems:
  - Diffie-Hellman key exchange protocol
  - U.S. Digital Signature Standard

Message Authentication

27

- Make sure Bob gets the message unaltered
- Don’t let Alice deny sending the message



- Don't care about eavesdropper Darth, unless Darth changes the message
- How can cryptography help?

DIGITAL SIGNATURES

28

Run public key system in reverse

- Only your private key can be used to write messages that will be decrypted by your public key
- Messages are not (necessarily) secret, but
  - know who sent them
  - know they haven’t been altered
- Generic use
  - unalterable, authenticated documents
  - critical counterweight to ease of digital editing
  - may even include time stamps
  - better than handwritten signature?

KEY MANAGEMENT WITH PUBLIC KEY ENCRYPTION

- Bob can send public key over insecure communication channel
- But how do you know Darth didn't send you his key instead?

KEY MANAGEMENT IS STILL HARD

- Still need to distribute public keys
  - **Ask recipient?**
    - Bad guys could intercept message and give her a bogus key
  - **Publish public key list in New York Times**
    - Bad guys could forge a New York Times, just for you
  - **Rely on a trusted network**
- More complications
  - **what if you don't know recipient?**
  - **what if sender and receiver are computers?**
- No escaping need for trust
  - **Rely on institutions, not technology**
  - **But at least now only need ONE trusted party**

A CENTRAL KEY DISTRIBUTOR

31

- Alice asks the distributor for Bob's public key
- Distributor sends key to Alice and "digitally signs" it
- Alice knows the key came from the distributor
  - Now just have to be sure that the distributor is honest and got Bob's key from Bob, not Darth
- Requires one secure communication per user
  - Bob sends public key to distributor when he joins the system
- Secret keys require secure communication between every pair of users

KEY ESCROW AND KEY RECOVERY

32

- What if key(s) are lost?
- What if an employee is away, gets fired, leaves for a competitor?
- What if the government wants to listen in?
  - legal wiretaps
  - espionage
- Key Escrow and Recovery Systems allow to access encrypted information without the proper key
  - like a Master key or a locksmith
  - encryption only as secure as the escrow/recovery procedures

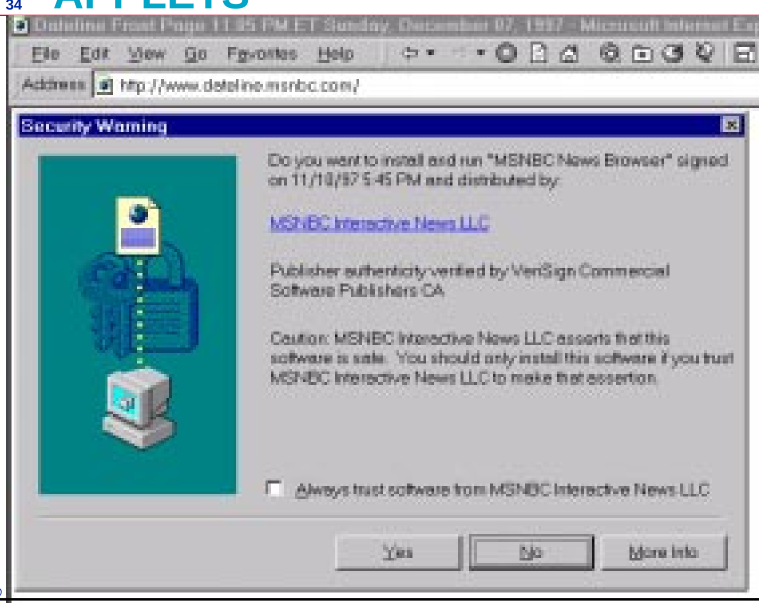
APPLICATIONS OF CRYPTOGRAPHY

33

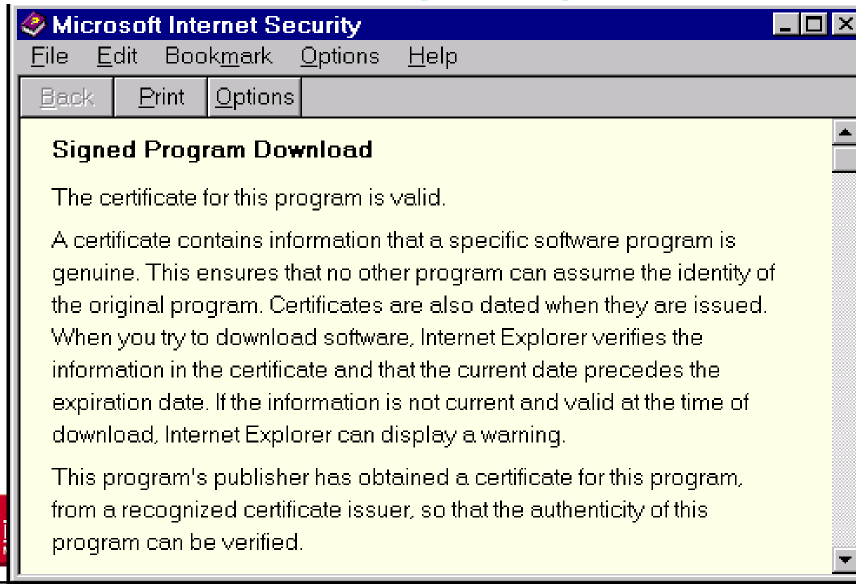
- Secure EDI
- Electronic Cash
  - verifiable, yet anonymous
  - smart cards or net cash
- Secure communications
  - email
  - telephones & faxes
- Tamper-proof documents
  - driver's licenses
  - designs & plans
  - checks & contracts

CERTIFIED SOFTWARE APPLETS

34



35 CERTIFIED SOFTWARE APPLETS (Cont'd)



36 WHY ISN'T CRYPTOGRAPHY MORE WIDELY USED?

- User ignorance & apathy
- User confusion
- Lack of interoperability
- U.S. Government restricts use

ISSUES

37

- Strong encryption does not equal security!
  - Subtle flaws on homegrown systems (& implementations)
  - Non-random keys
  - The weak link (just ask Kevin Mitnick or the NSA)
- Is a world of perfect privacy a good idea?

APPLICATION #1: NETWORK SECURITY

38

- Client/server computing
  - User has client program running on one machine
  - Client program requests services that may be running on other machines
- Why control access to services?
  - Can allow open access to network, but not to all services
  - Different privileges to users of one service
  - Billing: usage based pricing

39

NETWORK ACCESS CONTROL TECHNIQUES

- **None**
  - **Local machine verifies user identity at login**
  - **Works when all local machines are secure**
- **Host verification**
  - **Service verifies that host has authority to allow logins**
  - **Can separate out secure from unsecured machines**
- **User verification**
  - **Service verifies user's identity**
    - Don't trust the host to check user's identity
  - **Doesn't require secure local machines**

40

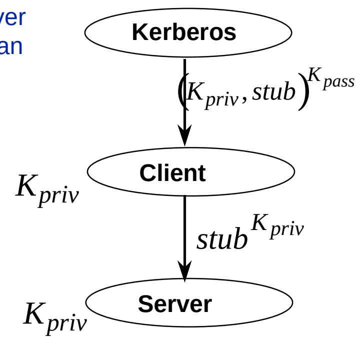
KERBEROS IDEA

- **User verification**
- **Kerberos knows user's password; servers (e.g., file servers) don't**
- **But don't send passwords over network**
  - **Not even encrypted passwords**
    - If bad guys capture encrypted password, they can replay it
- **Kerberos creates a "ticket" that's unusable unless the user types his password (locally)**

Kerberos Details

41

1. **Kerberos sends encrypted ticket to client**
2. **User types password to client so client can decrypt the ticket**
   - **Ticket has two parts**
     - a. A private key for talking to the file server
     - b. A ticket stub that only the file server can decrypt
3. **User sends a file request to the server**
   - **Request encrypted with the new key**
   - **Accompanied by stub**
4. **File server decrypts stub**
   - **Inside is another copy of the new key**
   - **File server decrypts the request**



APPLICATION #2: ELECTRONIC PAYMENTS

42

- **Model 1: encrypted credit card numbers**
  - **Actual payment is not electronic**
    - vendor collects from credit card company
  - **Used by Netscape**
- **Model 2: credit-debit instruments**
  - **Electronic signature on electronic check**
    - Vendor sends check to on-line bank
    - Bank verifies account
    - Bank transfers money from customer account to vendor account
  - **NetBill (CMU); NetCheque (USC)**

APPLICATION #2: ELECTRONIC PAYMENTS

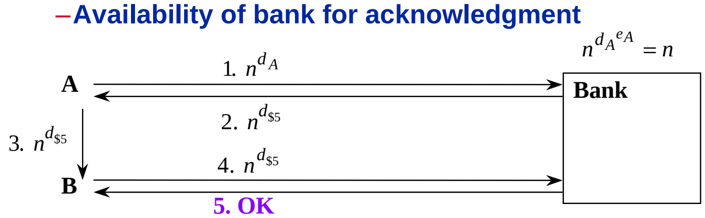
43

- **Model 3: electronic cash**
  - **User pays bank for "digitally signed" notes in advance**
  - **User transfers note to vendor**
  - **Vendor can cash it in at the bank**

EXAMPLE: DIGICASH

44

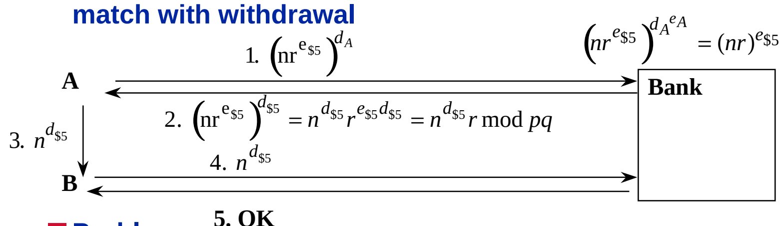
- **A withdraws $5 and stores it on her "smart-card"**
  - **A picks a large integer n, sends it (signed) to Bank**
  - **Bank sends back n, signed with its $5 signature**
    - Anyone can verify this signature
- **A gives the $5 to B**
  - **B verifies signature and asks Bank if money already spent**
- **Problems**
  - **Privacy: bank knows where and when A spent her money**
  - **Availability of bank for acknowledgment**



Example: Untraceable Currency

45

- Trick is to use "blind signatures"
  - Only note numbers n in a limited range are legitimate
- Alice multiplies note number n by a random factor r, unknown to bank
- Bank gets note back without random factor, so can't match with withdrawal



- Problems
  - May be easy to forge $n^{d_{\$5}}$ for some n in the right range, even if you can't forge for particular n
  - Still need to check with bank to prevent double spending

Detecting Duplicate Spending

46

- Don't require immediate clearance from bank
- B asks A a question before accepting money
  - A can't answer without knowing *r*, the blinding factor
  - A's answer does not reveal *r*
  - Answering two such questions does reveal *r*
    - Mechanism too complicated for us, but it works!
- No one but A can spend the money A withdrew from the bank
- If A spends it more than once, she reveals her identity, and the bank can track her down