Secrets & Lies, Knowledge & Trust. (Modern Cryptography)

COS 116
4/17/2008
Instructor: Sanjeev Arora

Cryptography: **1 :** secret writing

2 : the enciphering and deciphering of messages in secret code or cipher

- Ancient ideas: (pre-1976)
- Complexity-based cryptography (post-1976)

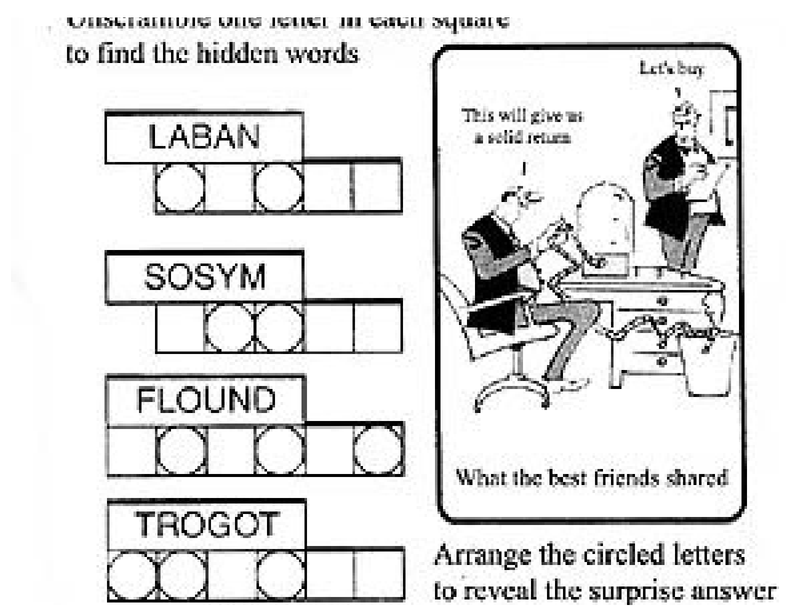
Basic component of Digital World; about much more than just encryption or secret writing.

Main themes of today's lecture

- Creating problems can be easier than solving them
- Difference between seeing information and making sense of it
- Role of randomness in the above
- Ability of 2 complete strangers to exchange secret information

Theme 1: Creating problems can be easier than solving them

Example:

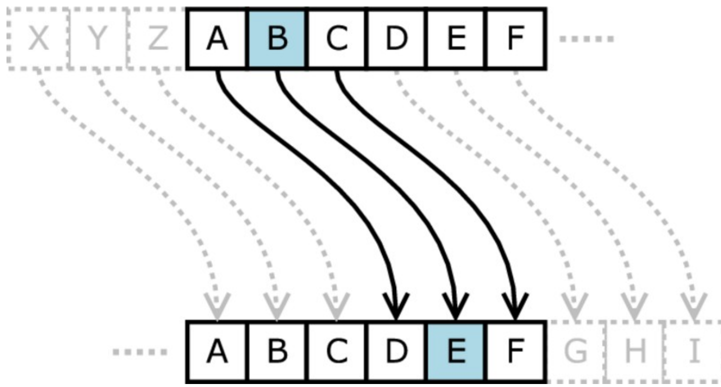


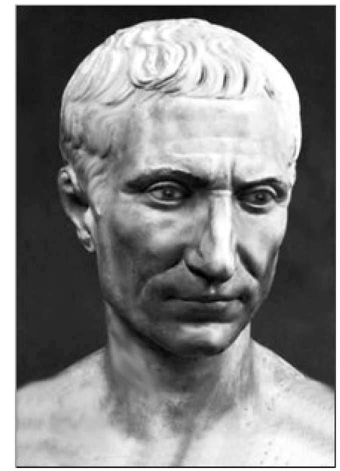
(Aside: This particular problem is trivial for computers!)

Reminiscent of something similar that is hard for current computers?

Letter scrambling: ancient cryptographic idea

Example 1: “Caesar cipher” (c. 100BC)





Example 2: Cipher used in conspiracy plot involving Queen Mary of Scots, 1587

From Discovery Channel, Apr 17 2006

Mafia Boss's Messages Deciphered

- “Boss of bosses” Bernardo Provenzano, captured after 40 years
- Sent “pizzini” (little messages on scraps of paper) using variant of Caesar cipher
- "...I met 512151522 191212154 and we agreed that we will see each other after the holidays...,“
- 5 = B, 12 = I, 15 = N, etc.

“It will keep your kid sister out, but it won't keep the police out.” - Bruce Schneier (Cryptographer)

Letter scrambling (cont.)

- Example 3: Enigma
  - Used by Nazi Germany (1940's)
  - Broken by British (Turing), Polish
  - "Won us the war." – Churchill

Moral: Computers → need for new ideas for encryption.

Integer factoring

Easy-to-generate problem

Hard to solve

Suggest an algorithm?
Running time?

- Say $n = 128$
- Generation:

  Pick two n-bit prime numbers p, q, and multiply them to get $r = pq$
- Factoring problem

  Given r, find p and q

Status of factoring

Despite many centuries of work, no efficient algorithms.

Believed to be computationally hard, but remains unproved ("almost –exponential time")

You rely on it every time you use e-commerce (coming up)

(Aside: If quantum computers ever get built, may become easy to solve.)

- Theme 2: Difference between seeing information and making sense of it

- Theme 3: Role of randomness.

Simple example that illustrates both: one-time pad ("daily codebook.")

Random source hypothesis

- Integral to modern cryptography

→ 0110101010011010011011101010010010001…

- I and my computer have a source of random bits
- These bits look completely random and unpredictable to the rest of the world.
- Ways to generate: Quantum phenomena in semi-conductors, timing between keystrokes, etc.

One-time pad (modern version)

- Goal: transmit *n*-bit message

- One-time pad: random sequence of *n* bits (*shared* between sender and receiver)

Using one-time pad

- Encryption: Interpret one-time pad as "noise" for the message
  - 0 means "don't flip"
  - 1 means "flip"

- Example:

Encryption

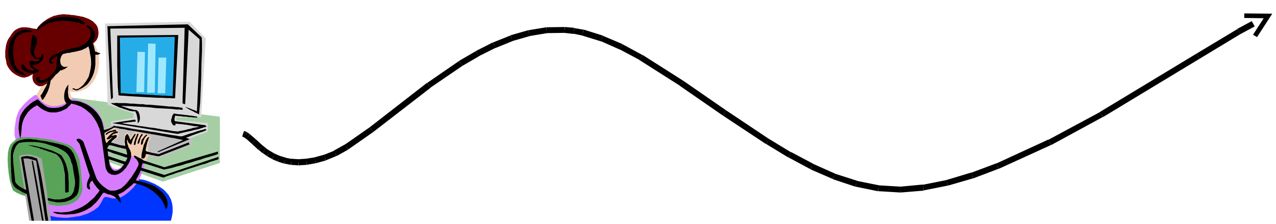
Message	0110010
Pad	1011001
Encrypted	1101011

Decryption

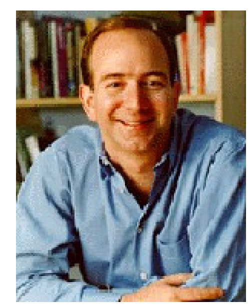
Encrypted	1101011
Pad	1011001
Message	0110010

Musings about one-time pad

- Incredibly strong security: encrypted message “looks random” – equally likely to be encryption of *any* n-bit string



amazon.com



Insecure link (Internet)

(Jeff Bezos ’86)

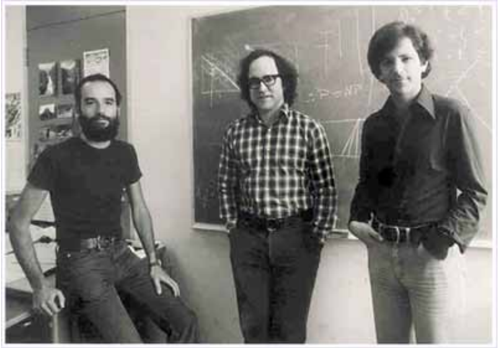
- How would you use one-time pad?
- How can you and Amazon agree on a one-time pad?

Theme: How perfect strangers can send each other encrypted messages.

Powerful idea: public-key encryption

- Diffie-Hellman-Merkle [1976]

- Rivest, Shamir, Adleman [1977]



Public-key cryptography

c = Encrypt(m, K_{pub})

amazon.com

Message m

Public key K_{pub}
(512 bit number, publicly available, e.g. from Verisign Inc)

Private key K_{priv}
(512-bit number, known only to Amazon.)

m = Decrypt(c, K_{priv})

- **Important**: encryption and decryption algorithms are *not* secret, only private key!

Public-key encryption at a conceptual level

- “Box that clicks shut, and only Amazon has the key to open it.”

- Example: Key exchange [Diffie-Hellman]
  - User generates random string (“one-time pad”)
  - Put it in box, ship it to Amazon
  - Amazon opens box, recovers random string

Public-Key Encryption at a mathematical level (RSA version)

Random Source Hypothesis!

Key generation: Pick random primes p, q.

Let N = p ¢ q

Find k that is not divisible by p, q. ("Public Key")

Encryption: m is encrypted as m^k (mod N)

Decryption: Symmmetric to Encryption; use "inverse" of k (this is private key)

"Modular" math

Last theme:

What does it mean to learn nothing?

Suggestions?

One-time pad revisited

c = 010010

Eve

m = 100100
pad = 110110

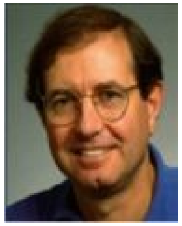
c = 010010
pad = 110110
m = 100100

- In what sense did Eve learn nothing about the message?
- Answer 1: Transmission was a sequence of random bits
- Answer 2: Transmission looked like something she could easily have generated herself

Eureka! moment for modern cryptography

Zero Knowledge Proofs

[Goldwasser, Micali, Rackoff '85]


Student

prox card

prox card reader

- Desire: Prox card reader should not store "signatures" – potential security leak
- Just ability to recognize signatures!
- Learn nothing about signature except that it **<u>is</u>** a signature

"ZK Proof": Everything that the verifier sees in the interaction, it could easily have generated itself.

Illustration: Zero-Knowledge Proof that "Sock A is different from sock B"

Sock A

Sock B

- Usual proof: "Look, sock A has a tiny hole and sock B doesn't!"
- ZKP: "OK, why don't you put both socks behind your back. Show me a random one, and I will say whether it is sock A or sock B. Repeat as many times as you like, I will always be right."
- Why does verifier learn "nothing"? (Except that socks are indeed different.)

Actual ZK Proofs

- Use numbers, number theory, etc.

(From Lecture 1): Public closed-ballot elections

- Hold an election in this room
  - Everyone can speak publicly (i.e. no computers, email, etc.)
  - At the end everyone must agree on who won and by what margin
  - No one should know which way anyone else voted
- Is this possible?
  - Yes! (A. Yao, Princeton)

"Privacy-preserving Computations" (Important research area)