Secrets & Lies, Knowledge & Trust.

(Modern Cryptography)

COS 116, Spring 2012

Adam Finkelstein

Cryptography

|kripˈtägrəfē| noun

the art of writing or solving codes.

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

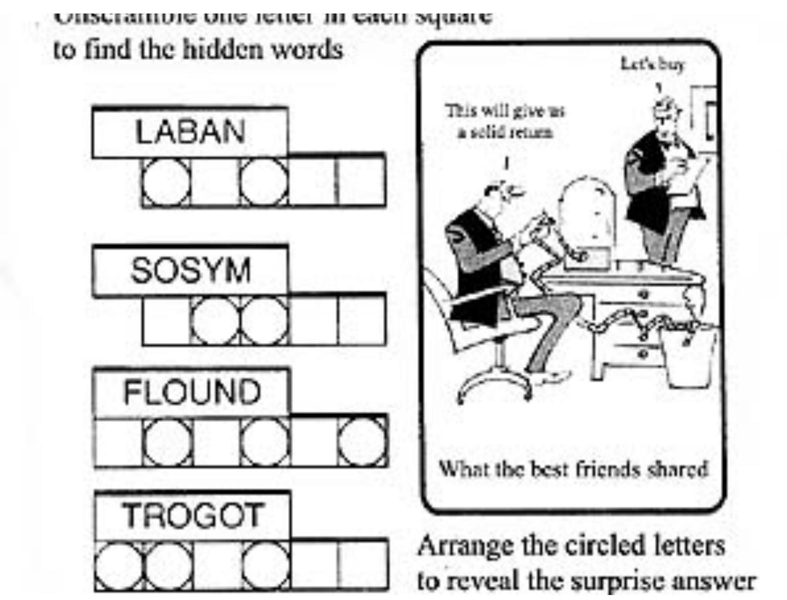
Basic component of Digital World.
More than just encryption / secret writing!

Main themes of today's lecture

- Creating problems can be easier than solving them
- Seeing information vs. making sense of it
- Role of randomness in the above
- Two complete strangers exchange secret information

Theme 1: Creating problems can be easier than solving them

Example:



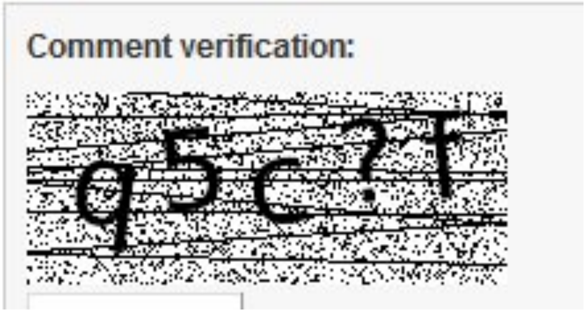
Unscramble one letter in each square to find the hidden words

LABAN
SOSYM
FLOUND
TROGOT
Let's buy
This will give us a solid return
What the best friends shared

Arrange the circled letters to reveal the surprise answer

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

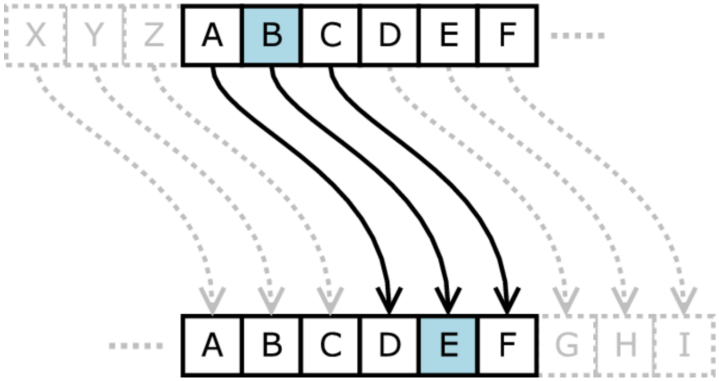
Reminiscent of something similar that is hard for current computers?

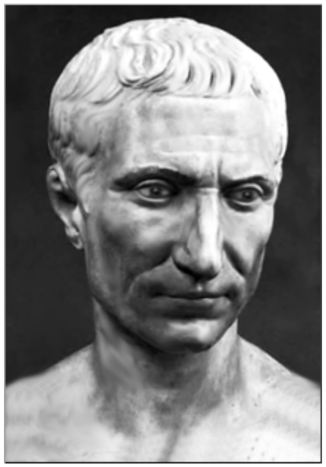


Comment verification:

Letter scrambling: ancient cryptographic idea

Example 1: “Caesar cipher” (c. 100BC)





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

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: Use of computer necessitates new ideas for encryption.

Integer factoring

Easy-to-generate problem

- Generation

  Pick two 32-digit prime numbers *p, q*, and multiply them to get $r = pq$

Hard to solve

- Factoring problem

  Given *r*: find *p* and *q*

We discussed an algorithm…
Running time?

Status of factoring

Despite many centuries of work, no efficient algorithms.

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

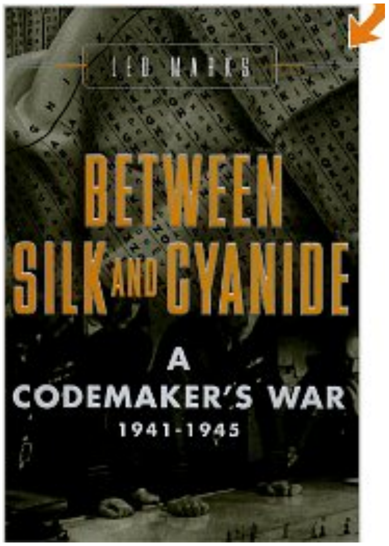
You rely on it every time you use e-commerce (coming up)

Note: If quantum computers ever get built, this may become easy to solve.

- Theme 2:
  Seeing information vs. 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…

- We have a source of random bits
- They look completely unpredictable
- Possible sources:
  Quantum phenomena,
  timing between keystrokes, etc.

One-time pad (modern version)

- Goal: transmit *n*-bit message

Alice

Bob

Eve

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

Using one-time pad

- Encryption:
  One-time pad scrambles the message, as follows:
  - 0 means “don't flip” the bit in the message
  - 1 means “flip” the bit in the message

- Example:

Encryption

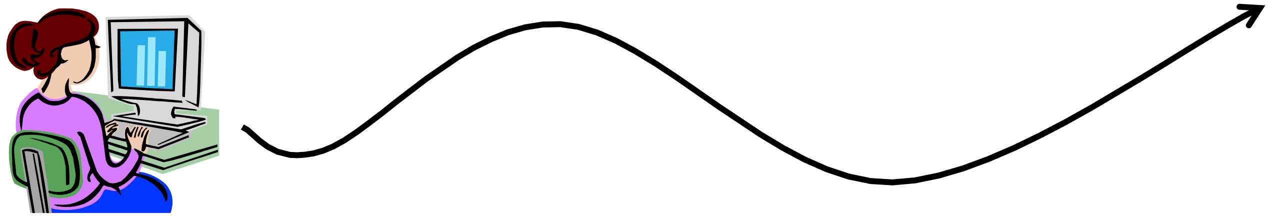
Message	0110010
Pad	1011001
Encrypted	1101011

Decryption (same rule!)

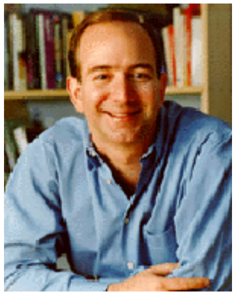
Encrypted	1101011
Pad	1011001
Message	0110010

Musings about one-time pad

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



Insecure link (Internet)



(Jeff Bezos '86)

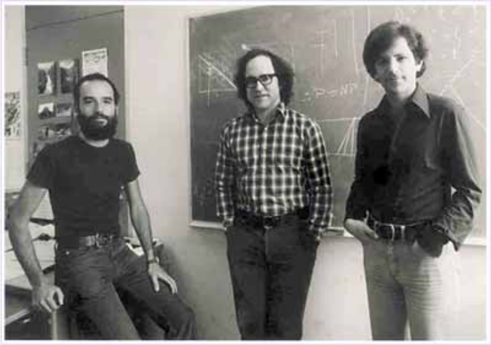
- How would you use one-time pad?
- How can you and Amazon agree on a one-time pad?

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)

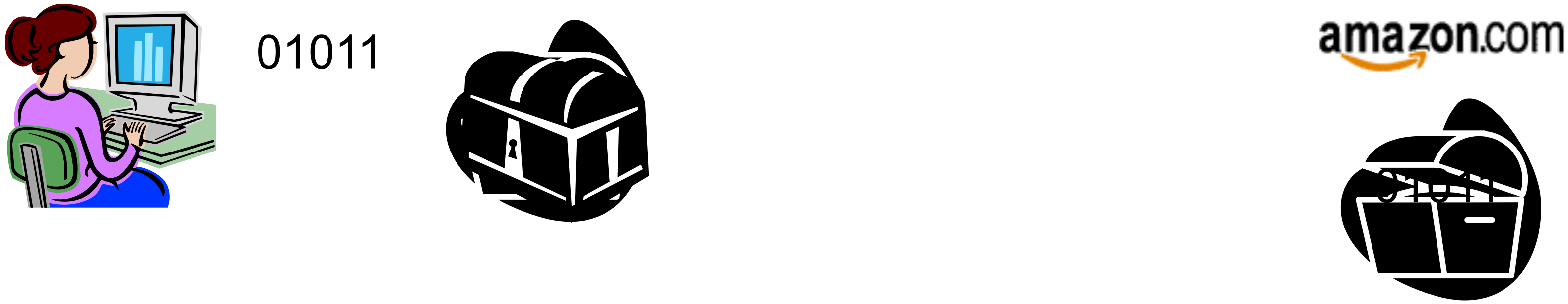
Private key K_{priv}
(512-bit number,
known only to
Amazon.)

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 $N = p \times q$

Primes and “modular” math

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)

(don’t sweat the details on this!)

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

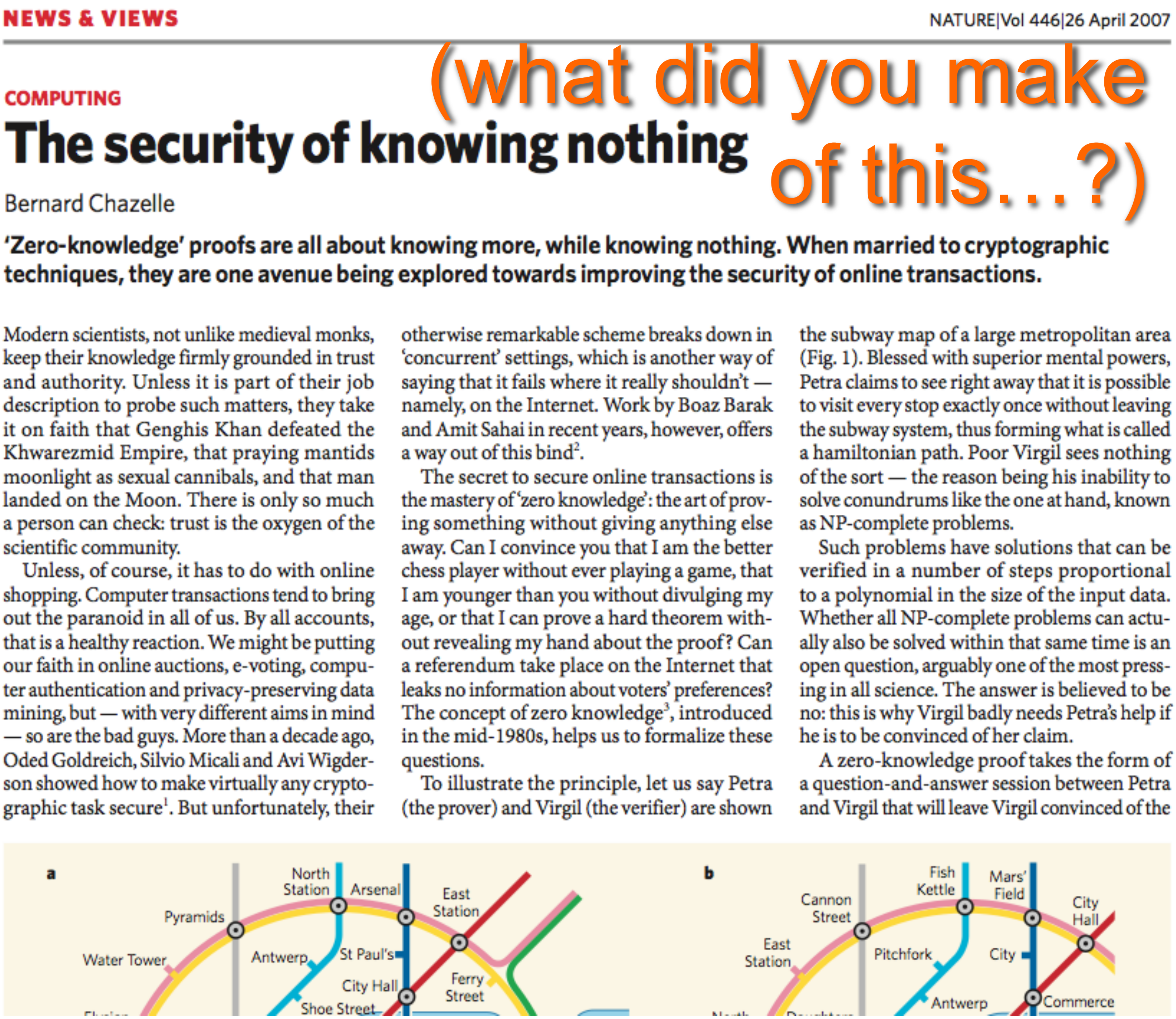
“ZK Proof”: Everything that the verifier sees in the interaction it could easily have generated itself.

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.)



NEWS & VIEWS

NATURE|Vol 446|26 April 2007

COMPUTING

The security of knowing nothing

(what did you make of this…?)

Bernard Chazelle

'Zero-knowledge' proofs are all about knowing more, while knowing nothing. When married to cryptographic techniques, they are one avenue being explored towards improving the security of online transactions.

Modern scientists, not unlike medieval monks, keep their knowledge firmly grounded in trust and authority. Unless it is part of their job description to probe such matters, they take it on faith that Genghis Khan defeated the Khwarezmid Empire, that praying mantids moonlight as sexual cannibals, and that man landed on the Moon. There is only so much a person can check: trust is the oxygen of the scientific community.

Unless, of course, it has to do with online shopping. Computer transactions tend to bring out the paranoid in all of us. By all accounts, that is a healthy reaction. We might be putting our faith in online auctions, e-voting, computer authentication and privacy-preserving data mining, but — with very different aims in mind — so are the bad guys. More than a decade ago, Oded Goldreich, Silvio Micali and Avi Wigderson showed how to make virtually any cryptographic task secure[1]. But unfortunately, their otherwise remarkable scheme breaks down in 'concurrent' settings, which is another way of saying that it fails where it really shouldn't — namely, on the Internet. Work by Boaz Barak and Amit Sahai in recent years, however, offers a way out of this bind[2].

The secret to secure online transactions is the mastery of 'zero knowledge': the art of proving something without giving anything else away. Can I convince you that I am the better chess player without ever playing a game, that I am younger than you without divulging my age, or that I can prove a hard theorem without revealing my hand about the proof? Can a referendum take place on the Internet that leaks no information about voters' preferences? The concept of zero knowledge[3], introduced in the mid-1980s, helps us to formalize these questions.

To illustrate the principle, let us say Petra (the prover) and Virgil (the verifier) are shown the subway map of a large metropolitan area (Fig. 1). Blessed with superior mental powers, Petra claims to see right away that it is possible to visit every stop exactly once without leaving the subway system, thus forming what is called a hamiltonian path. Poor Virgil sees nothing of the sort — the reason being his inability to solve conundrums like the one at hand, known as NP-complete problems.

Such problems have solutions that can be verified in a number of steps proportional to a polynomial in the size of the input data. Whether all NP-complete problems can actually also be solved within that same time is an open question, arguably one of the most pressing in all science. The answer is believed to be no: this is why Virgil badly needs Petra's help if he is to be convinced of her claim.

A zero-knowledge proof takes the form of a question-and-answer session between Petra and Virgil that will leave Virgil convinced of the

(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)