

Digital Envelopes, Zero Knowledge, and other wonders of modern cryptography



(How computational complexity
enables digital security & privacy)

Sanjeev Arora, Princeton University
& Center for Computational Intractability
(many slides from Avi Wigderson)

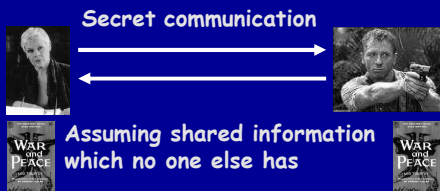
Cryptography: 1. secret writing

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

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

Modern crypto is about much more than just
encryption or secret writing.

Cryptography pre-1976 (before computational complexity)



Tasks	Traditional method
Encryption	Code books
Identification	Driver License
Money transfer	Notes, checks
Public bids	Sealed envelopes
Elections	Secret ballots

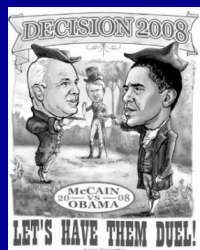


Need to be done online!

Qs. Why do you think this poses
a problem???

Example: 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)



What are we assuming
here???



Axiom 1: Agents are computationally limited.

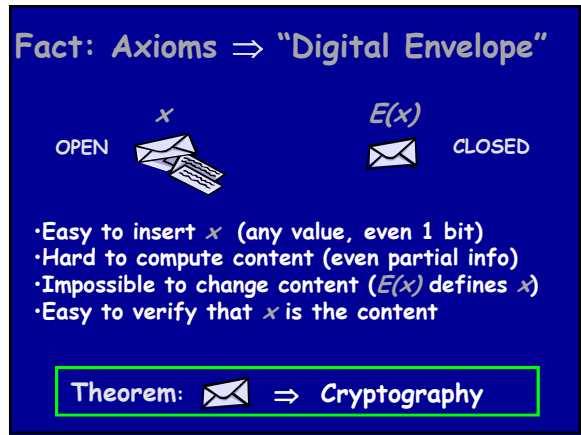
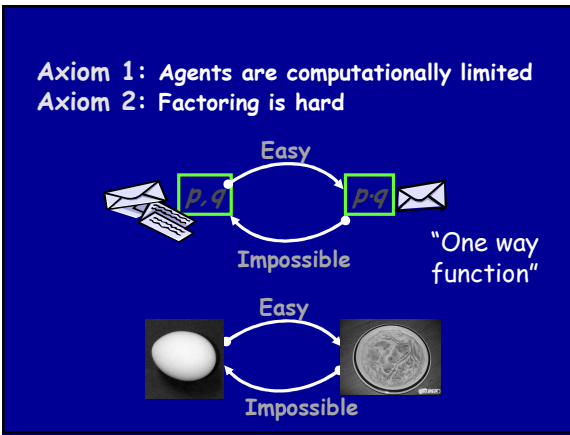
Consequence 1: Only tasks having efficient algorithms can be performed

Consequence 2: Can quantify when an agent learnt nothing ("Zero knowledge")

Recall: Creating Problems can be easier than solving them

Multiplication mult(23,67) = 1541	Factoring factor(1541) = (23,67)
grade school algorithm: n ² steps on n digit inputs	best known algorithm: exp(√n) steps on n digits
EASY Can be performed quickly for huge integers	HARD? We don't know! We'll assume it.

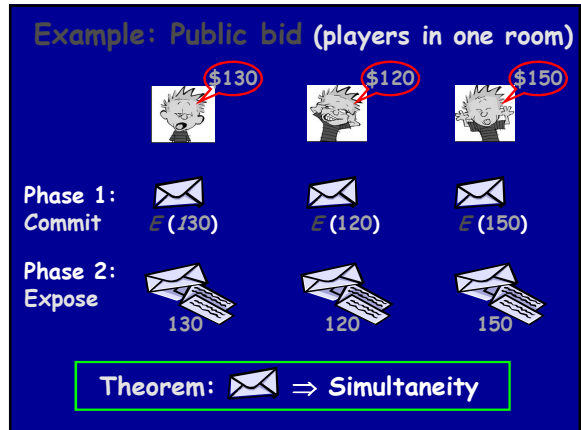
Axiom 2: Factoring is hard!



The power of the digital envelope

Examples of increasing difficulty

Mind games of the 1980's - before Internet & E-commerce were imagined



Blum
1981 **Public Lottery (on the phone)**



Alice: if  I get the car (else you do)


Bob: flipping..  What did you pick?

Theorem:  \Rightarrow Symmetry breaking


Identification / Passwords

Public password file

Name	$E(\text{pswd})$
...	...
alice	$P_{\text{alice}} = E(\dots)$
...	...
arora	$P_{\text{arora}} = E(\text{haha})$
...	...
bob	$P_{\text{bob}} = E(\dots)$
...	...



Computer: 1 checks if $E(\text{pswd}) = P_{\text{arora}}$
2 erases password from screen

~~Theorem:  Identification~~

Problem: Eavesdropping & repeated use!

Wishful thinking:
Computer should check if I know x such that $E(x) = P_{\text{arora}}$ without actually getting x

Zero-Knowledge Proof:


- Convincing
- Reveals no information

Aside: Copyrights

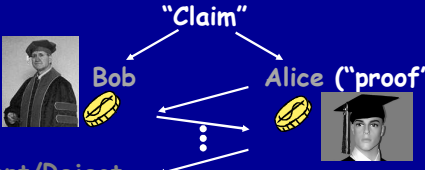
Dr. Alice: I can prove Riemann's Hypothesis

Prof. Bob: Impossible! What is the proof?

Dr. Alice: Lemma...Proof...Lemma...Proof...

Prof. Bob: Amazing!! I'll recommend tenure
 Amazing!! I'll publish first

Goldwasser-Micali
-Rackoff 1984 **Zero-Knowledge Proof**



Bob $\xrightarrow{\text{"Claim"}}$ Alice ("proof")


Accept/Reject


"Claim" true \rightarrow Bob accepts
Bob learns nothing


"Claim" false \rightarrow Bob rejects with high probability

Goldreich-Micali
-Wigderson 1986 **The universality of Zero-Knowledge**

Theorem: Everything you can prove at all, you can prove in Zero-Knowledge

Theorem [GMW]:  + short proof
 \Rightarrow efficient ZK proof

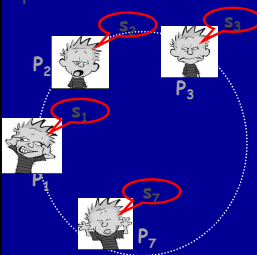


Theorem [GMW]:  \Rightarrow fault-tolerant protocols

s_i secret

Making any protocol fault-tolerant

1. P_2 send $m_1(s_2)$
2. P_7 send $m_2(s_7, m_1)$
3. P_1 send $m_3(s_1, m_1, m_2)$
- ...




Suppose that in step 1 P_2 sends X
 How do we know that $X=m_1(s_2)$?
 s_2 is a short proof of correctness!
 P_2 proves correctness in zero-knowledge!!

Some things we didn't have time for today

- RSA public-key cryptosystem and digital signature method
- Yao's computation-scrambling idea
- Various subtle security attacks (chosen ciphertext, chosen plaintext, etc. etc.) and how to guard against them
- Easier and speedier implementations of the zero knowledge idea using modular arithmetic (e.g. Fiat-Shamir identification)

Example: 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)




Requires Yao's circuit-scrambling ideas in addition to digital envelope

Example: Private communication

Alice and Bob want to have a completely private conversation.

They share no private information



Solved using RSA cryptosystem (in conjunction with signature authorities like Verisign)

Summary

Practically every cryptographic task can be performed securely & privately
 Assuming that players are computationally bounded and Factoring is hard.

- Computational complexity is essential!
- Hard problems can be useful!
- The theory predated (& enabled) the Internet
- What if factoring is easy (note: believed not to be NP-complete)?
- We have (very) few alternatives.

Major open question: Can cryptography be based on NP-complete problems ?