Special Topics in Cryptography

Mohammad Mahmoody

Last time

- RSA public key encryption
- Digital signatures

Today

- Finishing digital signatures
- Zero Knowledge Proofs
- Secure Computation.

Public Key Authentication: Digital Signatures

- Secure authentication without shared secret keys!

Defining Digital Signatures

	Enc	Auth
Priv	enc oracle	Mac(·) oracle
Pub	No need	still need sign oracle

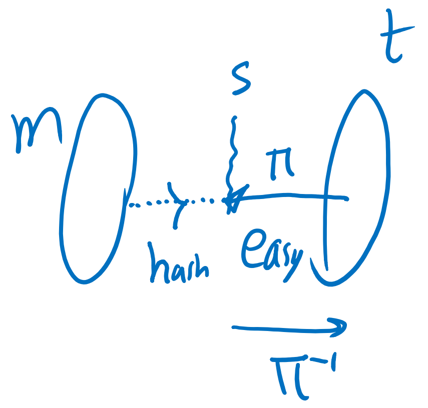
- Alice has a signing key sk and a verification key vk
- Using sk Alice can sign m with $\sigma = \text{Sign}_{sk}(m)$
- If Bob verifies $\text{Verif}_{vk}(m, \sigma) = 1$ he can be sure Alice signed m

- Security: For any poly-time adversary A who has access to a signing oracle $\text{Sign}_{sk}(\cdot)$, the probability of A finding (m, σ) for m not asked by A that also passes the test $\text{Verif}_{vk}(m, \sigma) = 1$ is negligible.

One possible idea based on TDPs (e.g. RSA)

π public $\quad$ π^{-1} secret

- Signing key: "private key" (or the trapdoor)
- Verification key: "public key" (or the description of the permutation)
- To sign m publish $\overset{\pi^{-1}}{\sigma}(m) = t = \pi^{-1}(m)$
- To verify (m, t) accept if and only if: $\pi(t) = m$



- Is it secure signature?
  No, because we can choose t first and then find m for it easily!

Sign

"Hash and sign" using ideal hash function

- Directly works for any message (arbitrary length) :

- Suppose $h\ :\{0,1\}^* \to \{0,1\}^n$ for security parameter n
- And we have trapdoor permutation π, π^{-1} on domain $\{0,1\}^n$

$$m \xrightarrow{h} s \underset{\pi}{\overset{\pi^{-1}}{\rightleftarrows}} t$$

- Signing key π^{-1}
- Verification key π
- To sign m, first get $s = h(m)$ and then output $\cancel{\sigma}\, t = \pi^{-1}(\cancel{m}\, s)$

Verif (m, t) : $h(m) \underset{s}{=} \pi(t) \longrightarrow$ output 1

$h(m) \neq \pi(t) \longrightarrow$ output 0 secure.

Why is it a good signing method?

if (π, π^{-1}) is a secure TDP

Thm: if h is a random function

and if A is an adv runs in poly(n) time

(and so it can only compute $h(\cdot)$ on poly(n) points).

$\rightarrow$ Pr[A winning in sec game against the "hash & sign" scheme] $\leq neg(\cdot)$

Proof: reduction: assuming such A

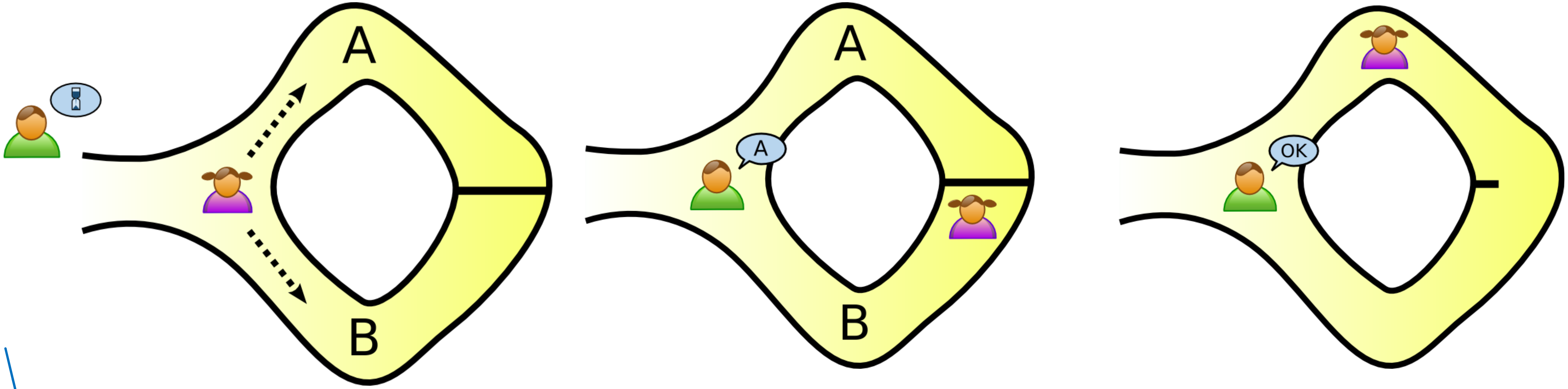
$\rightarrow$ turn it into B breaks TDP (π, π^{-1})

Zero Knowledge Proofs

- Proving the truth of statements, while revealing nothing about the proof!

Can we ever prove we know something without revealing the details of the secret?

- Alice knows a magic word to open the door inside the cave:



- How can she prove to Bob that she knows that word?

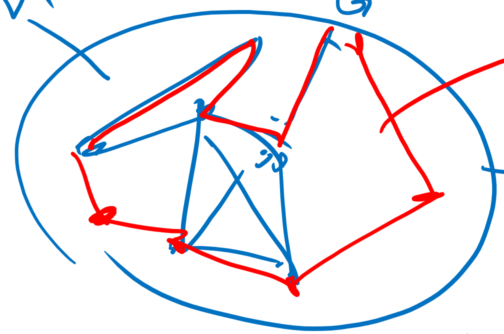
① Alice opens the door ② Bob enters and asks A/B
B ③ Alice exits from C

C

What is an "efficiently provable" property?

- Examples: graph G

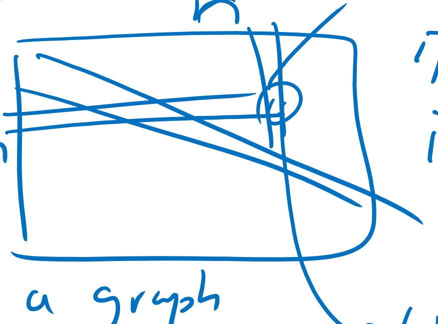
n G



Hamiltonian cycle

$i = M_{ij}$ iff $i \leftrightarrow j$

M_G n



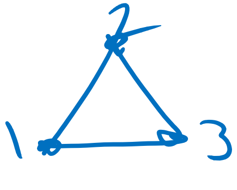
0/1

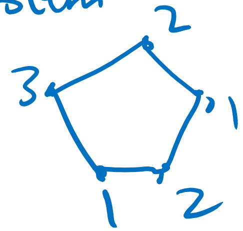
① No known poly time alg for ~~de~~ finding HC in a graph or even test if it exists

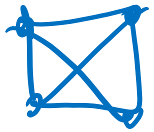
→ ② If somebody A knows cycle C in G.

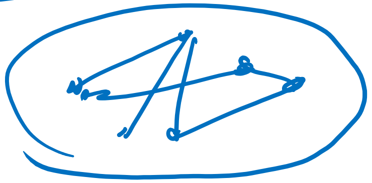
→ it is easy to convince others B

3-Coloring Problem: G









Yes: if ∃ way to color "nodes" with {1,2,3} such that

$i \leftrightarrow j$ -- $c(i) \neq c(j)$

No:

What is an "efficiently provable" property?

$L = \{\text{set of strings}\}$

- Complexity Class NP: set of all languages L where there is an **efficient** verifier V for proving membership in L. Name for all $x \in L$ there is a "witness" (or proof) w that $V(x, w) = 1$ and if $x \notin L$ then is No

witness making $V(x, w) = 1$

$L : \{ y \mid \exists x \quad \underbrace{h}_{\text{SHA256}}(x) = y \}$ $y \in L$, easy to prove using $\underline{x}$

$\longrightarrow L \in NP$

$\forall y \xrightarrow[\text{polytime}]{T} T(y) : \underset{G}{\text{graph}}$ $y \in L$ iff G is 3-Colorable

NP complete problems

L is NP complete if

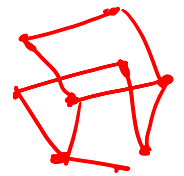
Solving L in poly-time can be used "in a very simple way" to solve <u>any</u> other $L' \in NP$.

GMW: membership in any NP language can be proved Zero Knowledge!

- Enough to do it for one NP complete problem only.
- Idea: using *interaction*
- Suppose we have digital lockable envelopes

A Prover

G:



B Verifier

Claim: $\boxed{\exists\ c : \{1,2,\ldots,n\} \to \{1,2,3\}}$

Such that. $\forall\ i,j$ $M(i,j)=1 \to c(i) \neq c(j)$

$x \longrightarrow Com[x] \longrightarrow B$

A y

①: Alice can "open" y into x and only x

② Bob has no idea what x is by looking at y

Proving a graph is 3 colorable

G: nodes 1, 2, ... n
edges
$e(i,j) = 1$ iff $i \leftrightarrow j$

P A Prov — Bob V Veri

$E_1, E_i \; E_n$

E_i: envelope containing $c(i)$

knows $c(i)$ $\forall i \in \{1 \ldots n\}$ such that $c(i) \neq c(j)$ if $i \leftrightarrow j$

Alice permutes the colors randomly

$\xleftarrow{i,j}$ Pick (i,j): connected edge at random among $e_1, e_2, \ldots e_m$

$i \leftrightarrow j$ random

open E_i E_j → c_i c_j

$c_i, c_j \in \{1, 2, 3\}$
$c_i \neq c_j$
accept

if C is Not a 3-coloring → Bob catches the Alice by prob? $\geq \frac{1}{m}$

$Pr[\text{Not catching Alice}] \leq (1 - \frac{1}{m})$

Why is this a convincing (sound) **interactive** proof?

because if G is not 3-colorable Bob catches Alice $\geq \frac{1}{m}$.

if we repeat protocol (from the beginning). k times

$$\Pr[\text{not catching}] \leq \left(1-\frac{1}{m}\right)^k$$

$K = 100\,m$

$$\leq \left(\left(1-\frac{1}{m}\right)^m\right)^{100} \leq e^{-100} \leq 2^{-100}$$

Why is this proof carrying “zero knowledge”?

in one interaction : $\exists$ simulator that efficiently <u>generates</u> what Bob observes.

Sim : ~~Gener~~ Pick (i,j) at random

choose random $c(i) \neq c(j)$

choose $c(k)$ so for all other nodes $k \neq i$, $k \neq j$

Put $c(1)$ —— $c(n)$ all in envelopes

only open E_i , E_j

Formal Definition of Zero Knowledge Proofs

$\boxed{\text{Sound}}$ ~~if~~ : if $G \notin L \rightarrow$

$\Pr\{V \text{ accept}\} \leq \text{neg}(n)$.

$\boxed{\text{Zero-knowlege}}$ $\exists$ poly-time Sim. $\forall\, G \in L$ (it is)

$\text{Sim}(G) \xrightarrow{\text{output}} T' \approx T$

T': fake transcipt. $\approx$: indist. T: actual transcript

How to get a lockable digital envelope?

Commitment Scheme.

Wrong $\underbrace{\text{Enc}(0\; b)}_{E} \approx \text{Enc}_k(b)$

open ~~En~~ open by sending key k

2 prop $\begin{cases} \text{hiding } \checkmark \\ \text{binding: } \exists \text{ at most one } b \text{ that we can open to.} \end{cases}$

Good: $h(b, \text{rand}) = E$ $\begin{cases} \text{hiding because } h \text{ is PRG} \\ \text{binding bec. } h \text{ is collision resistant.} \end{cases}$