Access Control and Privacy Policies (6)

Email:	christian.urban at kcl.ac.uk
Office:	S1.27 (1st floor Strand Building)
Slides:	KEATS (also homework is there)

Checking Solutions

How can you check somebody's solution without revealing the solution?

Checking Solutions

How can you check somebody's solution without revealing the solution?

Alice and Bob solve crosswords. Alice knows the answer for 21D (folio) but doesn't want to tell Bob.

You use an English dictionary:

- folio

Checking Solutions

How can you check somebody's solution without revealing the solution?

Alice and Bob solve crosswords. Alice knows the answer for 21D (folio) but doesn't want to tell Bob.

You use an English dictionary:

- folio

  "an individual leaf of paper or parchment, either loose as one of a series or forming part of a bound volume, which is numbered on the recto or front side only."

Checking Solutions

How can you check somebody's solution without revealing the solution?

Alice and Bob solve crosswords. Alice knows the answer for 21D (folio) but doesn't want to tell Bob.

You use an English dictionary:

- folio $\xrightarrow{1}$ individual

  "a single human being as distinct from a group"

Checking Solutions

How can you check somebody's solution without revealing the solution?

Alice and Bob solve crosswords. Alice knows the answer for 21D (folio) but doesn't want to tell Bob.

You use an English dictionary:

- folio $\xrightarrow{1}$ individual $\xrightarrow{2}$ human
  "relating to or characteristic of humankind"

Checking Solutions

How can you check somebody's solution without revealing the solution?

Alice and Bob solve crosswords. Alice knows the answer for 21D (folio) but doesn't want to tell Bob.

You use an English dictionary:

- folio $\xrightarrow{1}$ individual $\xrightarrow{2}$ human $\xrightarrow{3}$ or ...

Checking Solutions

How can you check somebody's solution without revealing the solution?

Alice and Bob solve crosswords. Alice knows the answer for 21D (folio) but doesn't want to tell Bob.

You use an English dictionary:

- folio $\xrightarrow{1}$ individual $\xrightarrow{2}$ human $\xrightarrow{3}$ or ...

this is essentially a hash function...but Bob can only check once he has also found the solution

Zero-Knowledge Proofs

Two remarkable properties of Zero-Knowledge Proofs:

- Alice only reveals the fact that she knows a secret, not the secret itself (meaning she can convince Bob that she knows the secret).
- Having been convinced, Bob cannot use the evidence in order to convince Carol that Alice knows the secret.

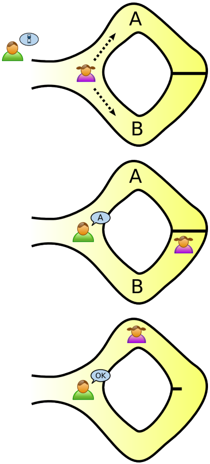
The Idea

The Alibaba cave:

1.

2.

3.



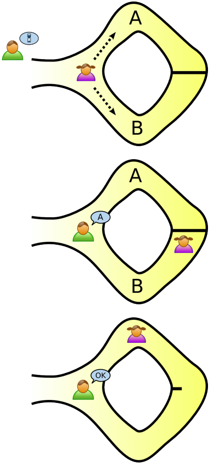
The Idea

The Alibaba cave:

1.

2.

3.



Even if Bob has a hidden camera, a recording will not be convincing to anyone else (Alice and Bob could have made it all up).

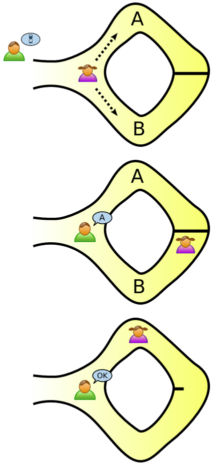
The Idea

The Alibaba cave:

1.

2.

3.



Even worse, an observer present at the experiment would not be convinced.

Applications of ZKPs

- authentication, where one party wants to prove its identity to a second party via some secret information, but doesn't want the second party to learn anything about this secret

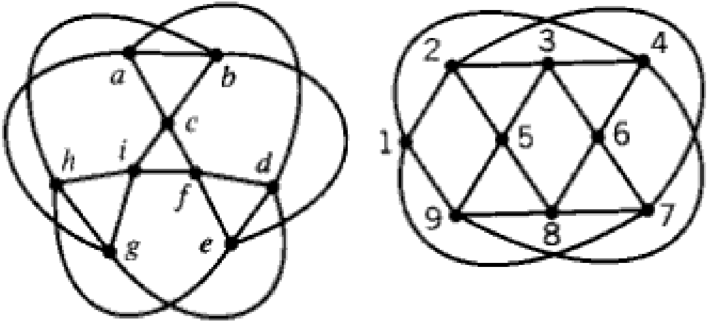
- to enforce honest behaviour while maintaining privacy: the idea is to force a user to prove, using a zero-knowledge proof, that its behaviour is correct according to the protocol

Central Properties

Zero-knowledge proof protocols should satisfy:

- **Completeness** If Alice knows the secret, Bob accepts Alice "proof" for sure.
- **Soundness** If Alice does not know the secret, Bob accepts her "proof" with a very small probability.

Graph Isomorphism



Finding an isomorphism between two graphs is an NP complete problem.

Graph Isomorphism Protoco

Alice starts with knowing an isomorphism σ between graphs G_1 and G_2

1. Alice generates an isomorphic graph H which she sends to Bob
2. Bob asks either for an isomorphism between G_1 and H, or G_2 and H
3. Alice and Bob repeat this procedure n times

Graph Isomorphism Protoco

Alice starts with knowing an isomorphism σ between graphs G_1 and G_2

1. Alice generates an isomorphic graph H which she sends to Bob
2. Bob asks either for an isomorphism between G_1 and H, or G_2 and H
3. Alice and Bob repeat this procedure n times

these are called commitment algorithms

Graph Isomorphism Protoco

If Alice knows the isomorphism, she can always calculate σ.

If she doesn't, she can only correctly respond if Bob's choice of index is the same as the one she used to form H. The probability of this happening is $\frac{1}{2}$, so after n rounds the probability of her always responding correctly is only $\frac{1}{2}^n$.

Graph Isomorphism Protoco

Why is the GI-protocol zero-knowledge?

Graph Isomorphism Protoco

Why is the GI-protocol zero-knowledge?

A: We can generate a fake transcript of a conversation, which cannot be distinguished from a “real” conversation.

Anything Bob can compute using the information obtained from the transcript can be computed using only a forged transcript and therefore participation in such a communication does not increase Bob’s capability to perform any computation.

Non-Interactive ZKPs

This is amazing: Alison can publish some data that contains no data about her secret, but this data can be used to convince anyone of the secret's existence.

Non-Interactive ZKPs (2)

Alice starts with knowing an isomorphism σ between graphs G_1 and G_2

1. Alice generates n isomorphic graphs $H_{1..n}$ which she makes public
2. she feeds the $H_{1..n}$ into a hashing function (she has no control over what the output will be)
3. Alice takes the first n bits of the output: whenever output is 0, she shows an isomorphism with G_1 ; for 1 she shows an isomorphism with G_2

Problems of ZKPs

- “grand chess master problem”
  (person in the middle)

- Alice can have multiple identities; once she committed a fraud she stops using one

Other Methods for ZKPs

Essentially every NP-problem can be used for ZKPs

- modular logarithms: Alice chooses public A, B, p; and private x

$$A^x \equiv B \; mod \; p$$

Commitment Stage

1. Alice generates z random numbers $r_1, \ldots, r_z$, all less than $p - 1$.
2. Alice sends Bob for all $1..z$

$$h_i = A^{r_i} \; mod \; p$$

3. Bob generates random bits $b_1, \ldots, b_z$ by flipping a coin
4. For each bit b_i, Alice sends Bob an s_i where

$$\begin{array}{ll} b_i = 0: & s_i = r_i \\ b_i = 1: & s_i = (r_i - r_j) \; mod \; (p - 1) \end{array}$$

where r_j is the lowest j where $b_j = 1$

Confirmation Stage

- For each b_i Bob checks whether s_i conforms to the protocol

$$\begin{array}{ll} b_i = 0: & A^{s_i} \equiv B \; mod \; p \\ b_i = 1: & A^{s_i} \equiv h_i * h_j^{-1} \; mod \; p \end{array}$$

Bob was send

$$r_j - r_j, r_m - r_j, \ldots, r_p - r_j \;\; \text{mod} \;\; \text{p}$$

where the corresponding bits were 1; Bob does not know r_j, he does not know any r_i where the bit was 1

Proving Stage

1. Alice proves she knows x, the discrete log of B she sends

$$s_{z+1} = (x - r_j)$$

2. Bob confirms

$$A^{s_{z+1}} \equiv B * h_j^{-1} \ mod\ p$$

Proving Stage

1. Alice proves she knows x, the discrete log of B she sends

$$s_{z+1} = (x - r_j)$$

2. Bob confirms

$$A^{s_{z+1}} \equiv B * h_j^{-1} \; mod \; p$$

In order to cheat, Alice has to guess all bits in advance. She has only 1 to 2^z chance.
(explanation → `http://goo.gl/irL9GK`)

Random Number Generators

- Computers are deterministic. How do they generate random numbers?

Random Number Generators

- Computers are deterministic. How do they generate random numbers?

- The most popular method to generate random numbers between 0 and m is: choose three integers

a	multiplier
c	increment
X_0	start value

and calculate

$$X_{n+1} = (a * X_n + c) \; mod \; m$$

Random Number Generator

$m =$	16	16
$X_0 =$	1	1
$a =$	5	5
$c =$	1	0

- Computers are deterministic. H
  generate random numbers?

- The most popular method to ge
  numbers between 0 and m is: choose three integers

a	multiplier
c	increment
X_0	start value

and calculate

$$X_{n+1} = (a * X_n + c) \; mod \; m$$