A Brief Introduction to the Zero Knowledge Proof

Avijit Dutta

Post-Doctoral Scholar

TCG-CREST, IAI

An Example

Millennium Problems

Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

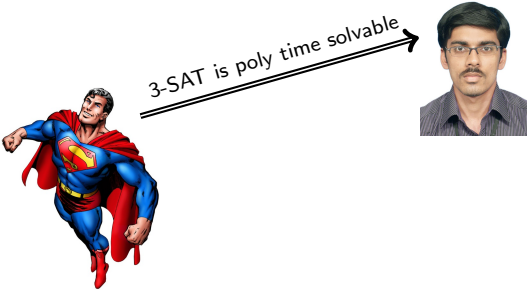
Riemann Hypothesis

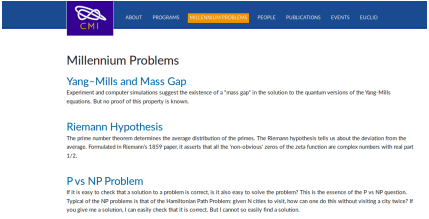
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

P vs NP Problem

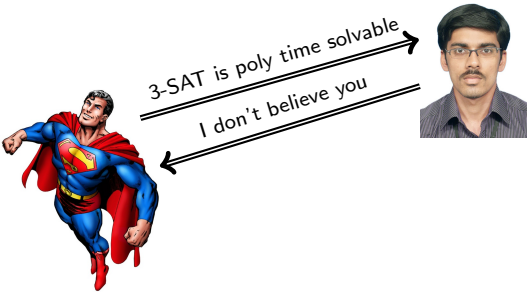
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

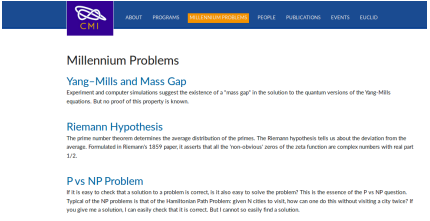
An Example


3-SAT is poly time solvable

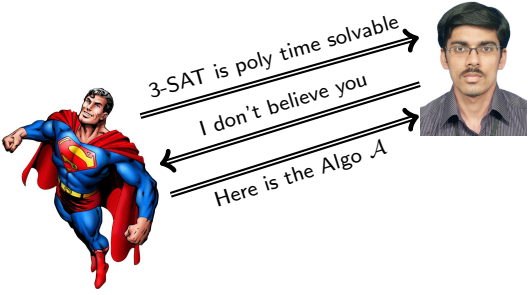

CMI
ABOUT
PROGRAMS
MILLENNIUM PROBLEMS
PEOPLE
PUBLICATIONS
EVENTS
EUCLID
Millennium Problems
Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.
Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.
P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

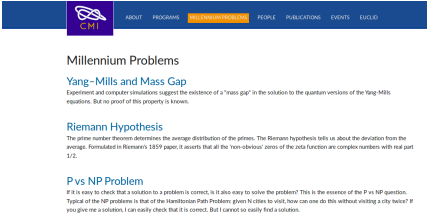
An Example


3-SAT is poly time solvable
I don't believe you

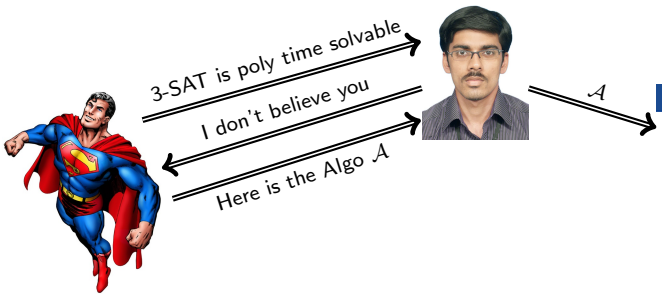

CMI
ABOUT
PROGRAMS
MILLENNIUM PROBLEMS
PEOPLE
PUBLICATIONS
EVENTS
EUCLID
Millennium Problems
Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.
Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.
P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

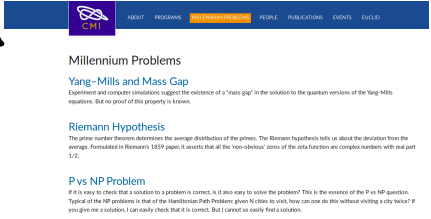
An Example


3-SAT is poly time solvable
I don't believe you
Here is the Algo $\mathcal{A}$

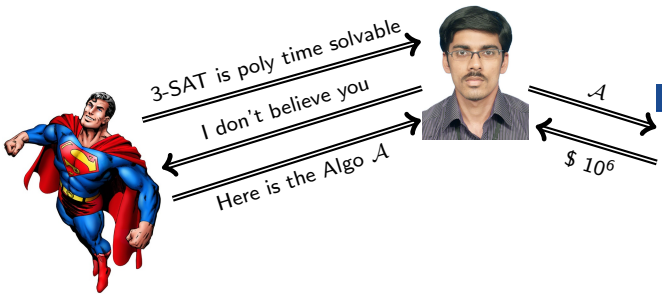

CMI
ABOUT
PROGRAMS
MILLENNIUM PROBLEMS
PEOPLE
PUBLICATIONS
EVENTS
EUCLID
Millennium Problems
Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.
Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.
P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

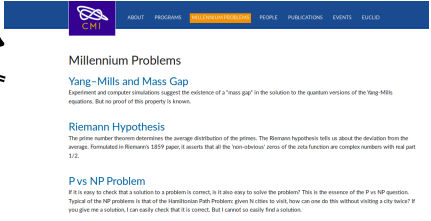
An Example


3-SAT is poly time solvable
I don't believe you
Here is the Algo $\mathcal{A}$
$\mathcal{A}$

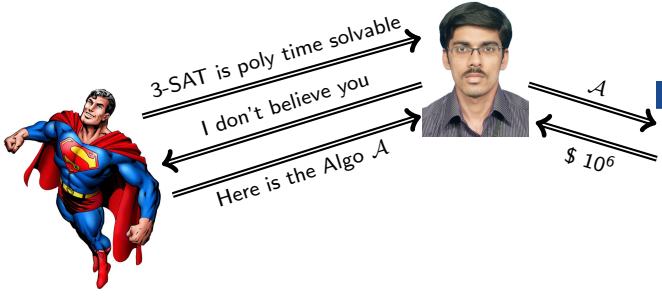

CMI
ABOUT
PROGRAMS
MILLENNIUM PROBLEMS
PEOPLE
PUBLICATIONS
EVENTS
EUCLID
Millennium Problems
Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.
Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.
P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

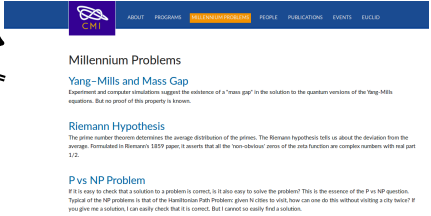
An Example


3-SAT is poly time solvable
I don't believe you
Here is the Algo $\mathcal{A}$
$\mathcal{A}$
$ 10^6


CMI
ABOUT
PROGRAMS
MILLENNIUM PROBLEMS
PEOPLE
PUBLICATIONS
EVENTS
EUCLID
Millennium Problems
Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.
Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.
P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

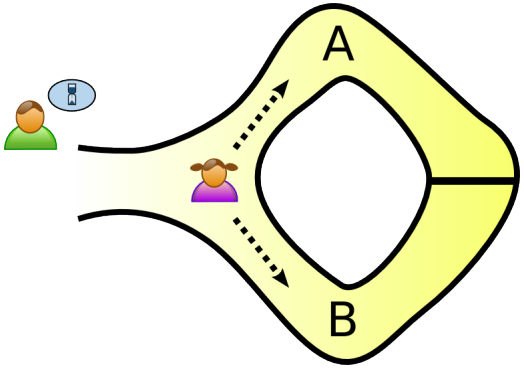
An Example



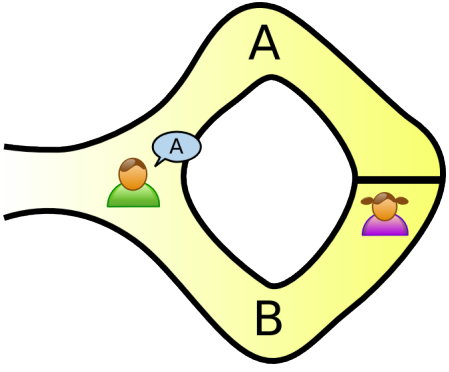


Can Superman convince me the validity of his statement without revealing the proof ?

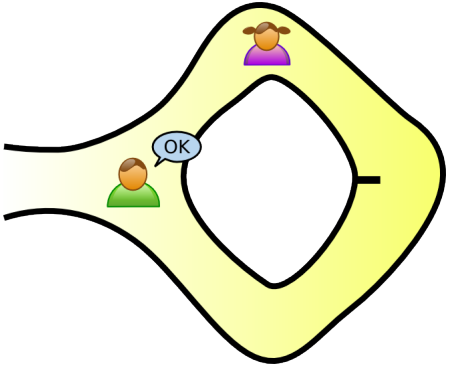
Motivation: The Story of Alibaba Cave



Motivation: The Story of Alibaba Cave



Motivation: The Story of Alibaba Cave



Breakthrough Result of GMR: Zero Knowledge Proof

- Conceived in 1985 by Shafi Goldwasser, Silvio Micali, and Charles Rackoff.
- Their paper "*The Knowledge Complexity of Interactive Proof-Systems*" got accepted in **SIAM'85**.
- Received **Godel Prize** in 1993 for advances in Theoretical Computer Science.

I. Proofs and Proof System

Notion of Proof

What is Proof ?

Notion of Proof

What is Proof ?

"A proof is whatever convinces me" – Shimon Even (1978)

Notion of Proof

What is Proof ?

"A proof is whatever convinces me" – Shimon Even (1978)

A proof involves two parties:

Notion of Proof

What is Proof ?

"A proof is whatever convinces me" – Shimon Even (1978)

A proof involves two parties:

- Prover – One who supplies the proof in favor of the statement

Notion of Proof

What is Proof ?

"A proof is whatever convinces me" – Shimon Even (1978)

A proof involves two parties:

- Prover – One who supplies the proof in favor of the statement
- Verifies – One who verifies the proof

Notion of Proof

What is Proof ?

"A proof is whatever convinces me" – Shimon Even (1978)

A proof involves two parties:

- Prover – One who supplies the proof in favor of the statement
- Verifies – One who verifies the proof

However, a proof makes no sense unless it can be efficiently verified / the verification process is simple.

Notion of Proof

What is Proof ?

"A proof is whatever convinces me" – Shimon Even (1978)

A proof involves two parties:

- Prover – One who supplies the proof in favor of the statement
- Verifies – One who verifies the proof

However, a proof makes no sense unless it can be efficiently verified / the verification process is simple.

- SAT has a simple verification procedure ($\in$ NP)
- $\overline{\text{SAT}}$ has no simple verification procedure ($\in$ co-NP)

Types of Proofs

Classical Proof

- The proof is fixed and written somewhere which is either
  - Self-evident, or
  - Derived from self-evident rules.
- These proofs are static in nature.
- Examples are mathematical proofs, proofs for NP-Languages.

Types of Proofs

Classical Proof

- The proof is fixed and written somewhere which is either
  - Self-evident, or
  - Derived from self-evident rules.
- These proofs are static in nature.
- Examples are mathematical proofs, proofs for NP-Languages.

Interactive Proof (IP)

- Proof involves exchanges of information in multiple rounds between prover and verifier
- Truth is established when the verifier accepts the hypothesis
- Proof is dynamic in nature.

A Formal Notion of Proof System

- $\mathcal{L} = \{(S, n) : S \text{ is a true mathematical stmt with a proof of length } \leq n\}$

A Formal Notion of Proof System

- $\mathcal{L} = \{(S, n) : S \text{ is a true mathematical stmt with a proof of length } \leq n\}$
- $\mathcal{L} \subseteq \{0,1\}^*$

A Formal Notion of Proof System

- $\mathcal{L} = \{(S, n) : S \text{ is a true mathematical stmt with a proof of length } \leq n\}$
- $\mathcal{L} \subseteq \{0,1\}^*$

Given $\mathcal{L}$ and $x \in \{0,1\}^*$, goal is to prove $x \in \mathcal{L}$.

A Formal Notion of Proof System

- $\mathcal{L} = \{(S, n) : S \text{ is a true mathematical stmt with a proof of length } \leq n\}$
- $\mathcal{L} \subseteq \{0,1\}^*$

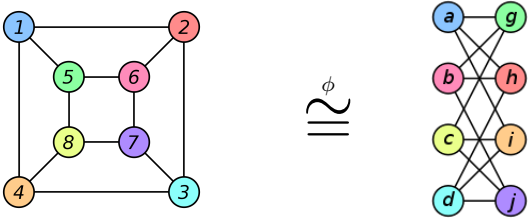
Given $\mathcal{L}$ and $x \in \{0,1\}^*$, goal is to prove $x \in \mathcal{L}$.

Valid Proof System

- Efficient- Verification of the proof should be simple (run time should be polynomial of the size of the proof)
- Completeness - True statement must have a proof ($\forall x \in \mathcal{L}, \exists \pi$ s.t. $V(x, \pi) = \top$)
- Soundness - False statement does not have any proof ($\forall x \notin \mathcal{L}, \forall \pi$ s.t. $V(x, \pi) = \bot$)

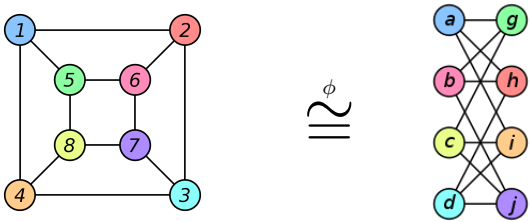
NP- An Example of a Classical Valid Proof System

Graph Isomorphism



NP- An Example of a Classical Valid Proof System

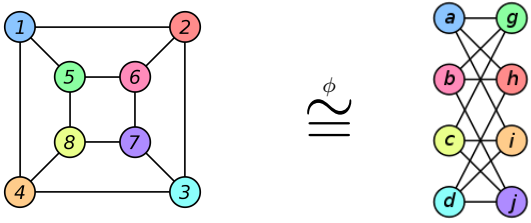
Graph Isomorphism



Isomorphism

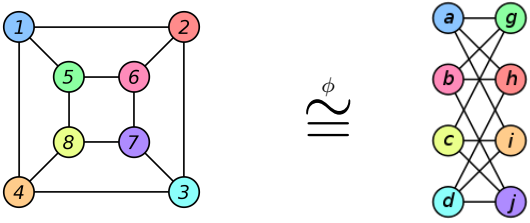
$$\{\phi(1) = a, \phi(2) = h, \phi(3) = d, \phi(4) = i, \phi(5) = g, \phi(6) = b, \phi(7) = j, \phi(8) = c\}$$

NP- An Example of a Classical Valid Proof System



- Prover (P) sends ϕ as the proof π to the verifier (V)
- V verifies π is a valid permutation. Run time is poly.
- Proof is complete and sound.

NP- An Example of a Classical Valid Proof System



- Prover (P) sends ϕ as the proof π to the verifier (V)
- V verifies π is a valid permutation. Run time is poly.
- Proof is complete and sound.

- NP is the set of languages with classical valid proof system
- No interaction between P and V

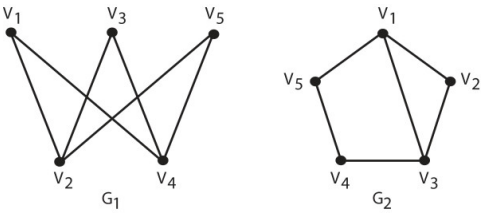
Example of a language not in NP

Can we have a language which cannot be proved in classical proof system ?

Example of a language not in NP

Can we have a language which cannot be proved in classical proof system ?

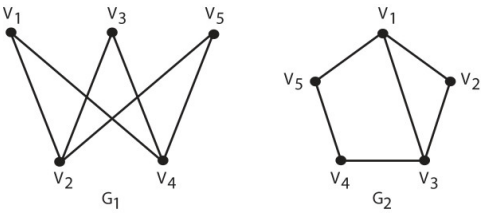
Graph Non-Isomorphism (GNI)



Example of a language not in NP

Can we have a language which cannot be proved in classical proof system ?

Graph Non-Isomorphism (GNI)

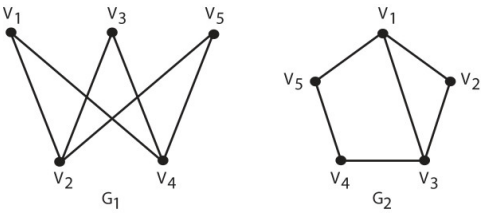


Send all possible permutations ϕ on G_1 to V

Example of a language not in NP

Can we have a language which cannot be proved in classical proof system ?

Graph Non-Isomorphism (GNI)

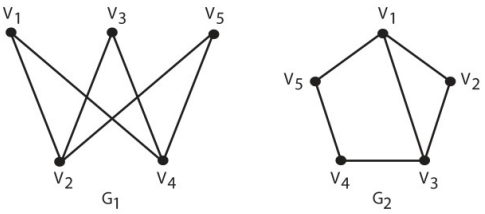


Send all possible permutations ϕ on G_1 to V – inefficient verification process

Example of a language not in NP

Can we have a language which cannot be proved in classical proof system ?

Graph Non-Isomorphism (GNI)



Send all possible permutations ϕ on G_1 to V – inefficient verification process

In fact, we do not know GNI $\in$ NP

Interactive Proof System (IP)

Interactive Protocol

- P wants to prove $x \in \mathcal{L}$
- V is probabilistic in nature – it tosses coins and hence errs with some probability – (Randomness)
- Instead of reading the proof, V $\leftrightarrow$ P to establish the validity of the assertion – (Interactive)
- V finally accepts the proof with some probability.

Interactive Proof System (IP)

Interactive Protocol

- P wants to prove $x \in \mathcal{L}$
- V is probabilistic in nature – it tosses coins and hence errs with some probability – (Randomness)
- Instead of reading the proof, V $\leftrightarrow$ P to establish the validity of the assertion – (Interactive)
- V finally accepts the proof with some probability.

Valid Interactive Proof System

- Efficient- Verification of the proof should be simple – (V is a ppt algorithm)
- Completeness - $\forall x \in \mathcal{L}$, Pr[V accepts in (P,V)$(x)$] $= 1$
- Soundness - $\forall x \notin \mathcal{L}, \forall$P*, Pr[V accepts in (P*, V)(x)] $\leq 1/2$

Interactive Proof System (IP)

Interactive Protocol

- P wants to prove $x \in \mathcal{L}$
- V is probabilistic in nature – it tosses coins and hence errs with some probability – (Randomness)
- Instead of reading the proof, V $\leftrightarrow$ P to establish the validity of the assertion – (Interactive)
- V finally accepts the proof with some probability.

Valid Interactive Proof System

- Efficient- Verification of the proof should be simple – (V is a ppt algorithm)
- Completeness - $\forall x \in \mathcal{L}$, Pr[V accepts in (P,V)$(x)$] $= 1$
- Soundness - $\forall x \notin \mathcal{L}, \forall$P*, Pr[V accepts in (P*, V)(x)] $\leq 1/2$

IP System is said to be "efficient" if P is given the witness

Distinguishing Problem – An Example of IP

P

V

I can distinguish the taste of Pepsi from Coke

Distinguishing Problem – An Example of IP

P

V

I can distinguish the taste of Pepsi from Coke

I don't believe you

Distinguishing Problem – An Example of IP

P

V

I can distinguish the taste of Pepsi from Coke

I don't believe you

How does P prove her claim to V ?

Distinguishing Problem – An Example of IP

P

V

Tell me is it Coke / Pepsi?

$b = 0$, pour pepsi
$b = 1$, pour coke

Distinguishing Problem – An Example of IP

P

V

Tell me is it Coke / Pepsi?

$b' \in \{0, 1\}$

$b = 0$, pour pepsi
$b = 1$, pour coke

Distinguishing Problem – An Example of IP

P

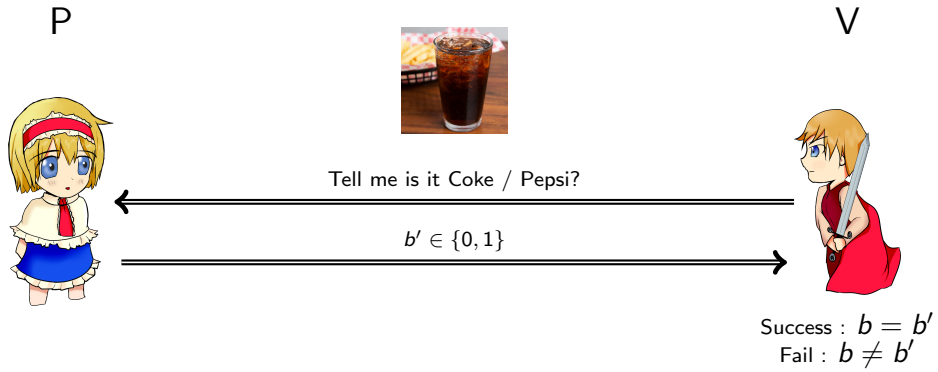
V

Tell me is it Coke / Pepsi?

$b' \in \{0, 1\}$

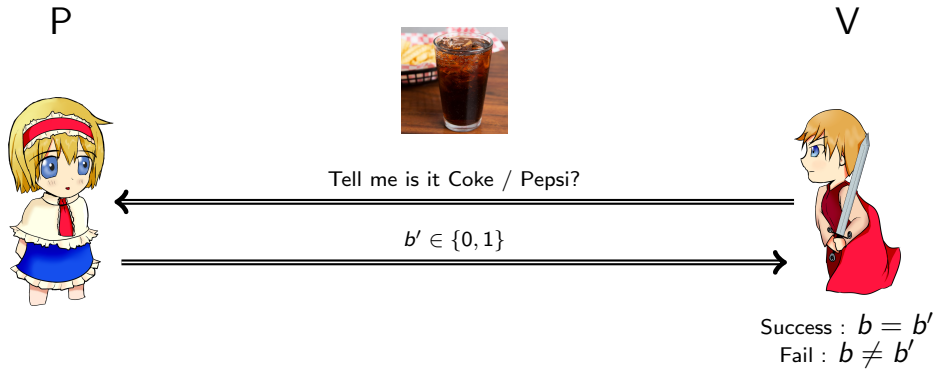
Success : $b = b'$
Fail : $b \neq b'$

Distinguishing Problem – An Example of IP



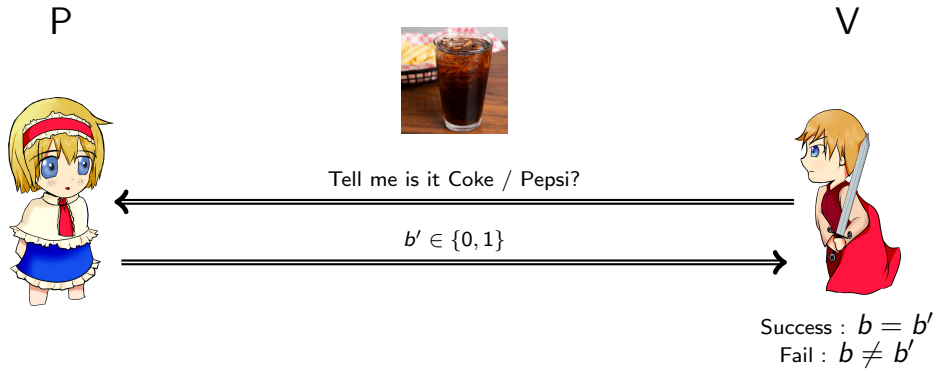
- If P really knows the difference then it always succeeds – **(Complete)**

Distinguishing Problem – An Example of IP



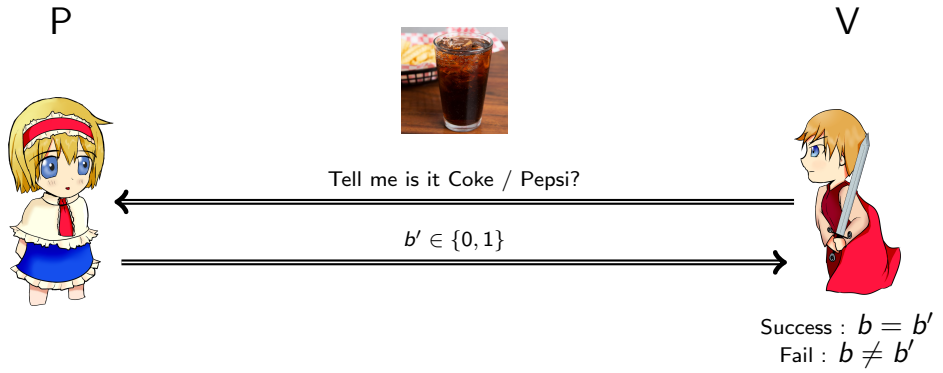
- If P really knows the difference then it always succeeds – **(Complete)**
- If P does not know, then it fails with probability (?)

Distinguishing Problem – An Example of IP



- If P really knows the difference then it always succeeds – **(Complete)**
- If P does not know, then it fails with probability (?) $1/2$

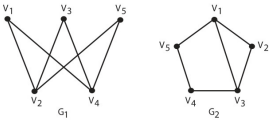
Distinguishing Problem – An Example of IP



- If P really knows the difference then it always succeeds – **(Complete)**
- If P does not know, then it fails with probability (?) $1/2$
- Repeat the expermient afresh and continues for t times – **Soundness error** 2^{-t}

GNI has an Interactive Proof

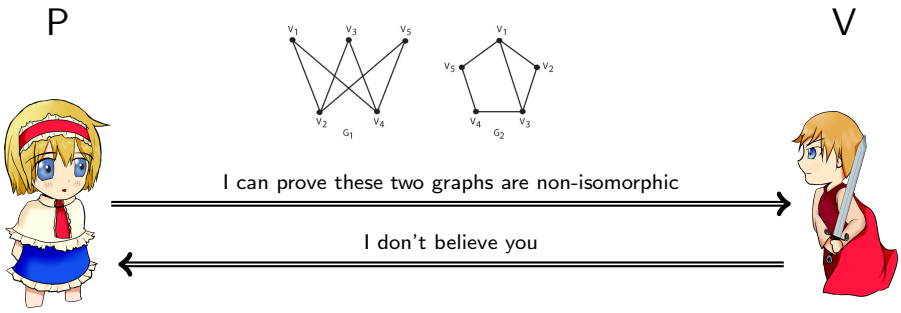
P



V

I can prove these two graphs are non-isomorphic

GNI has an Interactive Proof



GNI has an Interactive Proof

P

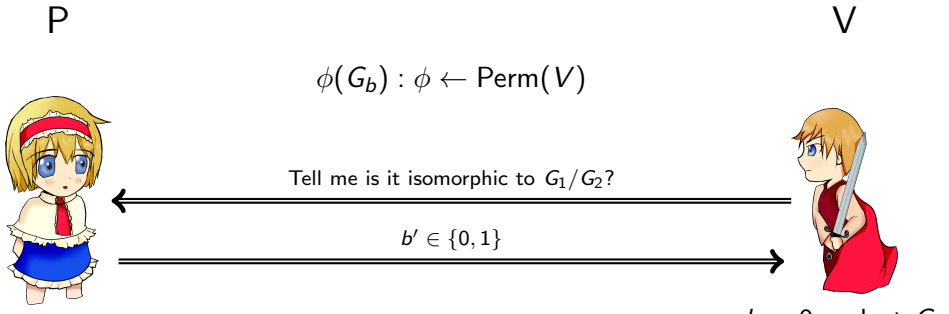
V

$\phi(G_b) : \phi \leftarrow \mathrm{Perm}(V)$

Tell me is it isomorphic to G_1/G_2?

$b = 0$, select G_1
$b = 1$, select G_2

GNI has an Interactive Proof



GNI has an Interactive Proof

P

V

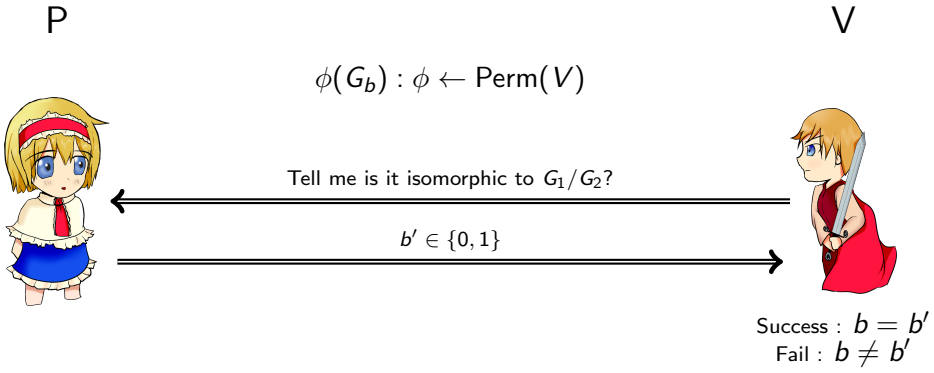
$\phi(G_b) : \phi \leftarrow \mathrm{Perm}(V)$

Tell me is it isomorphic to G_1/G_2?

$b' \in \{0, 1\}$

Success : $b = b'$
Fail : $b \neq b'$

GNI has an Interactive Proof



- Complete and Soundness error 2^{-t}
- Trival to see that NP $\subseteq$ IP
- IP = PSPACE

II. (Zero) Knowledge

The Notion of Knowledge

What is Knowledge ?

The Notion of Knowledge

What is Knowledge ?

Knowledge is the ability to complete a new task – Rafael Pass and Abhi Shelat

The Notion of Knowledge

What is Knowledge ?

Knowledge is the ability to complete a new task – Rafael Pass and Abhi Shelat

A conversation between two parties conveys knowledge when it allows the recipient to complete a "new" task that she could not complete before

The Notion of Knowledge

What is Knowledge ?

Knowledge is the ability to complete a new task – Rafael Pass and Abhi Shelat

A conversation between two parties conveys knowledge when it allows the recipient to complete a "new" task that she could not complete before

Can we quantify knowledge ?

The Notion of Knowledge

What is Knowledge ?

Knowledge is the ability to complete a new task – Rafael Pass and Abhi Shelat

A conversation between two parties conveys knowledge when it allows the recipient to complete a "new" task that she could not complete before

Can we quantify knowledge ?

- Hard to quantify knowledge.

The Notion of Knowledge

What is Knowledge ?

Knowledge is the ability to complete a new task – Rafael Pass and Abhi Shelat

A conversation between two parties conveys knowledge when it allows the recipient to complete a "new" task that she could not complete before

Can we quantify knowledge ?

- Hard to quantify knowledge.
- However, to quantify knowledge inherent in a message m, it is sufficient to quantify how much easier it becomes to compute a "new" function given m

The Notion of Knowledge– An Example

RSA Cryptosystem

- $N = pq, \phi(N) = (p-1)(q-1)$
- choose $e \in \mathbb{Z}_N^*$, $d = e^{-1} \mod \phi(N)$
- $pk_A = (N, e), sk_A = (N, d)$

A

B

$(m =$ "Meeting @ 10:30 p.m" $, \text{Sign}_{sk_A}(m))$

The Notion of Knowledge– An Example

I know the factorization of N

A

$(m =$ " Meeting @ 10:30 p.m" , $\text{Sign}_{sk_A}(m))$

B

The Notion of Knowledge– An Example

I know the factorization of N

I don't believe you

A

B

$(m =$ "Meeting @ 10:30 p.m" $, \mathrm{Sign}_{sk_A}(m))$

The Notion of Knowledge– An Example

I know the factorization of N

I don't believe you

(p, q)

Compute $sk_A = (N, d)$

A

B

$(m =$ "Meeting @ 10:30 p.m" $, \mathrm{Sign}_{sk_A}(m))$

The Notion of Knowledge– An Example

A

B

The Notion of Knowledge– An Example

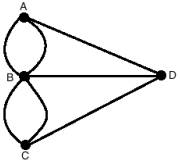
A

B

$(m =$ "You are fired" $, \mathrm{Sign}_{sk_B}(m))$

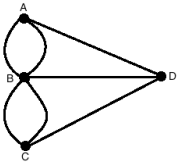
$(m =$ "You Bastard! Get out of this office" $, \mathrm{Sign}_{sk_A}(m))$

The Notion of Knowledge– An Example



"I know the graph is eulerian"

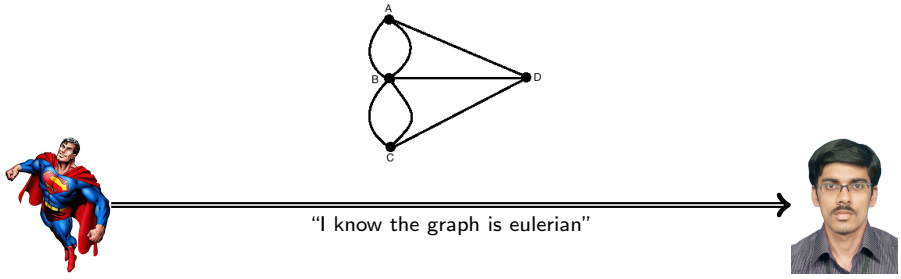
The Notion of Knowledge– An Example



"I know the graph is eulerian"

Does the message convey any knowledge ?

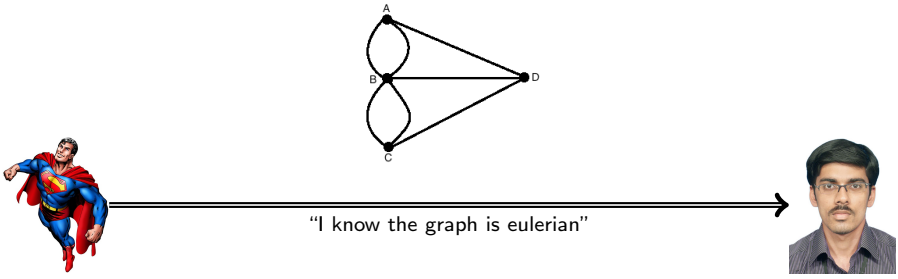
The Notion of Knowledge– An Example



Does the message convey any knowledge ?

No! I can very well compute whether the graph is "eulerian" or not.

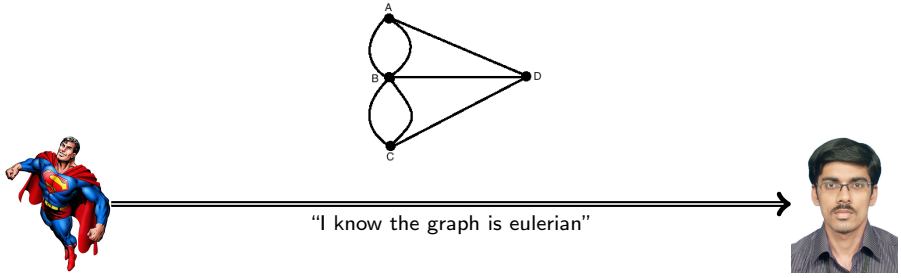
The Notion of Knowledge– An Example



Does the message convey any knowledge ?

No! I can very well compute whether the graph is "eulerian" or not. – (**Zero-Knowledge**)

The Notion of Knowledge– An Example



Does the message convey any knowledge ?

No! I can very well compute whether the graph is "eulerian" or not. – (**Zero-Knowledge**)

Information which conveys no knowledge is called **zero knowledge**

III. (I) + (II) $\Rightarrow$ Zero Knowledge Proof.

Notion of Zero Knowledge Proof (ZKP)

Can a classical proof system be a zero knowledge proof system ?

Notion of Zero Knowledge Proof (ZKP)

Can a classical proof system be a zero knowledge proof system ?

It must be an interactive proof system

- Efficient- Verification of the proof should be simple – (V is a ppt algorithm)
- Completeness - $\forall x \in \mathcal{L}, \Pr[\text{V accepts in (P,V)}(x)] = 1$
- Soundness - $\forall x \notin \mathcal{L}, \forall \text{P}^*, \Pr[\text{V accepts in } (\text{P}^*, \text{V})(x)] \leq 1/2$

Notion of Zero Knowledge Proof (ZKP)

Can a classical proof system be a zero knowledge proof system ?

It must be an interactive proof system

- Efficient- Verification of the proof should be simple – (V is a ppt algorithm)
- Completeness - $\forall x \in \mathcal{L}, \Pr[\text{V accepts in (P,V)}(x)] = 1$
- Soundness - $\forall x \notin \mathcal{L}, \forall \text{P}^*, \Pr[\text{V accepts in } (\text{P}^*, \text{V})(x)] \leq 1/2$

How do we model the proof system does not convey any knowledge ?

Notion of Simulator

- Simulator is a PPT Turing Machine

Notion of Simulator

- Simulator is a PPT Turing Machine
- Given only $x \in \mathcal{L}$ (and no access to P), can produce a transcript τ, that "looks similar" to the transcript that results from the interaction between the honest prover and the verifier.

Notion of Simulator

- Simulator is a PPT Turing Machine
- Given only $x \in \mathcal{L}$ (and no access to P), can produce a transcript τ, that "looks similar" to the transcript that results from the interaction between the honest prover and the verifier.

Definition of ZKP System

- (P, V) be an efficient IP System for a language $\mathcal{L}$.

Notion of Simulator

- Simulator is a PPT Turing Machine
- Given only $x \in \mathcal{L}$ (and no access to P), can produce a transcript τ, that "looks similar" to the transcript that results from the interaction between the honest prover and the verifier.

Definition of ZKP System

- (P, V) be an efficient IP System for a language $\mathcal{L}$.
- (P, V) is a ZKP system for $\mathcal{L}$, if $\forall$ PPT $V^*, \exists$ a PPT M^* such that $\forall x \in \mathcal{L}$

$$\boxed{(P, V^*)(x) \cong M^*(x)}$$

Notion of Simulator

- Simulator is a PPT Turing Machine
- Given only $x \in \mathcal{L}$ (and no access to P), can produce a transcript τ, that "looks similar" to the transcript that results from the interaction between the honest prover and the verifier.

Definition of ZKP System

- (P, V) be an efficient IP System for a language $\mathcal{L}$.
- (P, V) is a ZKP system for $\mathcal{L}$, if $\forall$ PPT $V^*, \exists$ a PPT M^* such that $\forall x \in \mathcal{L}$

$$\boxed{(P, V^*)(x) \cong M^*(x)}$$

$(P, V^*)(x)$ is the output of V^* after interacting with P on the common input $x \in \mathcal{L}$

Notion of Simulator

Rationale of Simulator

- It postulates that whatever a party can do "efficiently" by itself cannot be considered a gain from interaction with the outside.
- However, failure to provide a simulation of an interaction does NOT necessarily mean that this interaction results in some "real gain" (in some intuitive sense).
- What matters is that any "real gain" can NOT occur whenever we are able to present a simulation.

Notion of Simulator

Rationale of Simulator

- It postulates that whatever a party can do "efficiently" by itself cannot be considered a gain from interaction with the outside.
- However, failure to provide a simulation of an interaction does NOT necessarily mean that this interaction results in some "real gain" (in some intuitive sense).
- What matters is that any "real gain" can NOT occur whenever we are able to present a simulation.

Example: Eulerian Graph problem has a trivial ZKP

Variants of Zero Knowledge Proofs

Perfect Zero Knowledge Proof

(P, V) is PZK for $\mathcal{L}$ if $\forall \mathrm{V}^*$, $\exists M^*$ s.t. $\forall x \in \mathcal{L}$ the following two conditions hold:

- $Pr[M^\star(x) = \perp] \leq 1/2$,
- $\forall \alpha$, $\Pr[M^*(x) = \alpha \mid M^*(x) \neq \perp] = \Pr[(\mathrm{P}, \mathrm{V}^*)(x) = \alpha]$.

Variants of Zero Knowledge Proofs

Perfect Zero Knowledge Proof

(P, V) is PZK for $\mathcal{L}$ if $\forall \mathsf{V}^*$, $\exists M^*$ s.t. $\forall x \in \mathcal{L}$ the following two conditions hold:

- $Pr[M^*(x) = \perp] \leq 1/2$,
- $\forall \alpha$, $\Pr[M^*(x) = \alpha \mid M^*(x) \neq \perp] = \Pr[(\mathsf{P}, \mathsf{V}^*)(x) = \alpha]$.

Statistical Zero Knowledge

(P, V) is SZK for $\mathcal{L}$ if $\forall \mathsf{V}^*$, $\exists M^*$ s.t. $\forall x \in \mathcal{L}$ the following two conditions hold: ensembles are statistically close as functions of $|x|$:

- $Pr[M^*(x) = \perp] \leq 1/2$,
- $\{(\mathsf{P}, \mathsf{V}^*)(x)\}_{x \in \mathcal{L}} \approx_{sd} \{M^*(x)\}_{x \in \mathcal{L}}$.

Variants of Zero Knowledge Proof

Computational Zero Knowledge

(P, V) is CZK for $\mathcal{L}$ if $\forall V^*$, $\exists M^*$ s.t. $\forall x \in \mathcal{L}$ the following two conditions hold: are computationally indistinguishable:

- $Pr[M^*(x) = \perp] \leq 1/2$,
- $\{(P, V^*)(x)\}_{x \in \mathcal{L}} \approx_c \{M^*(x)\}_{x \in \mathcal{L}}$.

Variants of Zero Knowledge Proof

Computational Zero Knowledge

(P, V) is CZK for $\mathcal{L}$ if $\forall V^*$, $\exists M^*$ s.t. $\forall x \in \mathcal{L}$ the following two conditions hold: are computationally indistinguishable:

- $Pr[M^*(x) = \perp] \leq 1/2$,
- $\{(P, V^*)(x)\}_{x \in \mathcal{L}} \approx_c \{M^*(x)\}_{x \in \mathcal{L}}$.

Relation among the notions

Variants of Zero Knowledge Proof

Computational Zero Knowledge

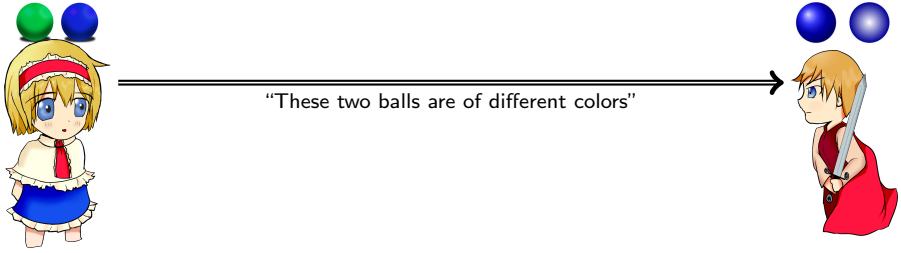
(P, V) is CZK for $\mathcal{L}$ if $\forall V^*$, $\exists M^*$ s.t. $\forall x \in \mathcal{L}$ the following two conditions hold: are computationally indistinguishable:

- $Pr[M^*(x) = \perp] \leq 1/2$,
- $\{(P, V^*)(x)\}_{x \in \mathcal{L}} \approx_c \{M^*(x)\}_{x \in \mathcal{L}}$.

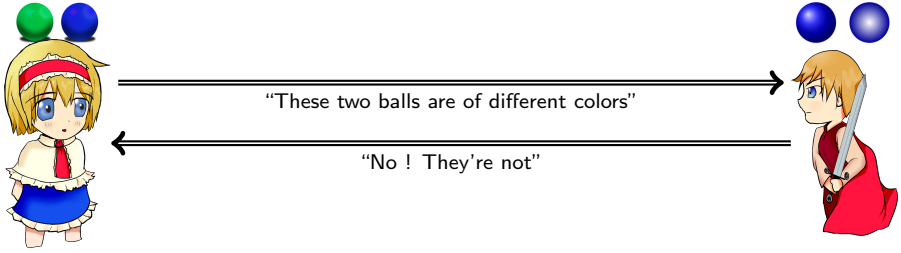
Relation among the notions

$$\boxed{BPP \subset PZK \subseteq SZK \subseteq CZK \subseteq IP}$$

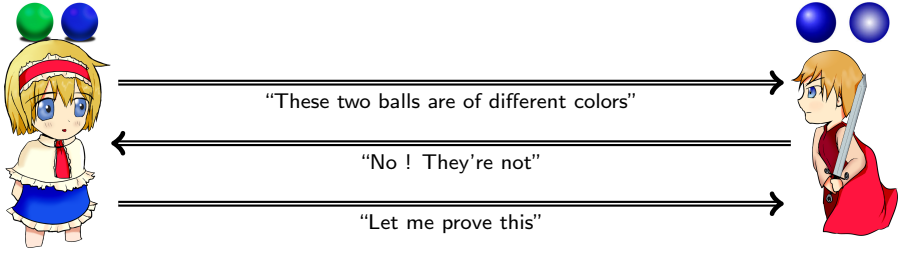
Examples of ZKP : Two Balls and the Color-Blind Friend



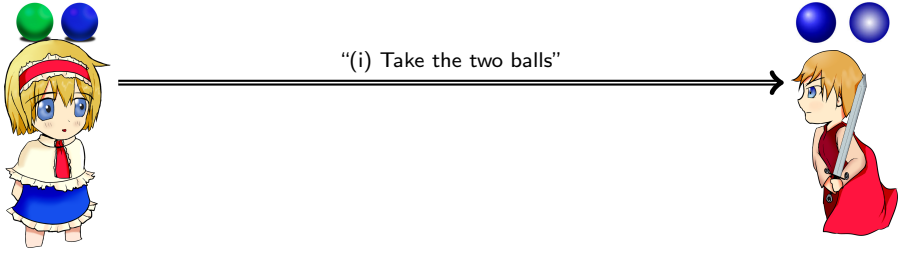
Examples of ZKP : Two Balls and the Color-Blind Friend



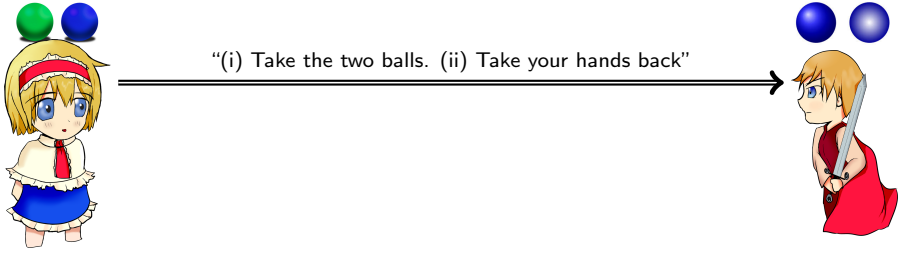
Examples of ZKP : Two Balls and the Color-Blind Friend



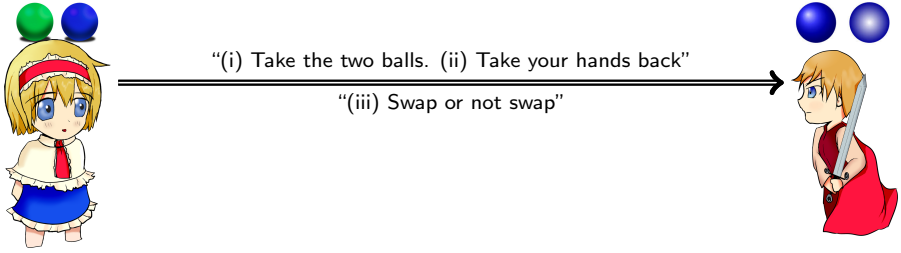
Examples of ZKP : Two Balls and the Color-Blind Friend



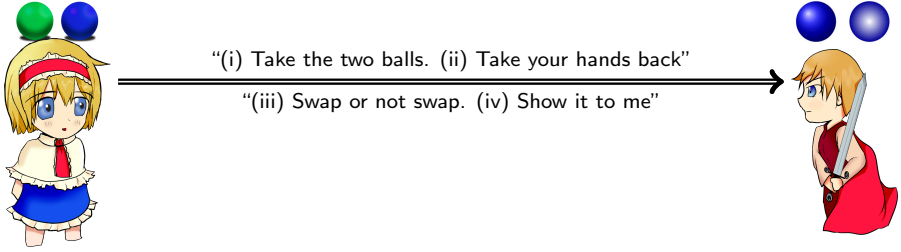
Examples of ZKP : Two Balls and the Color-Blind Friend



Examples of ZKP : Two Balls and the Color-Blind Friend



Examples of ZKP : Two Balls and the Color-Blind Friend



Examples of ZKP : Two Balls and the Color-Blind Friend

"(i) Take the two balls. (ii) Take your hands back"

"(iii) Swap or not swap. (iv) Show it to me"

swap if $b = 0$
not swap if $b = 1$

Examples of ZKP : Two Balls and the Color-Blind Friend

"(i) Take the two balls. (ii) Take your hands back"

"(iii) Swap or not swap. (iv) Show it to me"

"See my hands and tell whether I swapped or not"

swap if $b = 0$
not swap if $b = 1$

Examples of ZKP : Two Balls and the Color-Blind Friend

"(i) Take the two balls. (ii) Take your hands back"

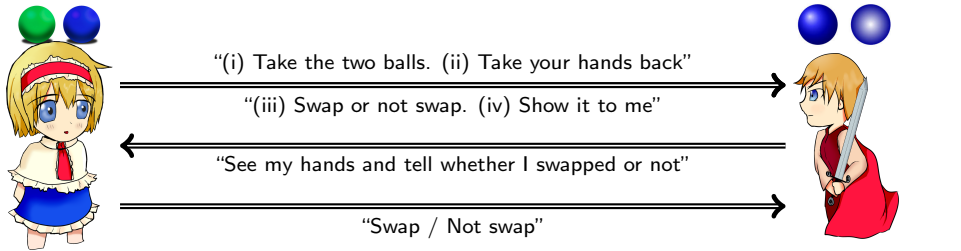
"(iii) Swap or not swap. (iv) Show it to me"

"See my hands and tell whether I swapped or not"

"Swap / Not swap"

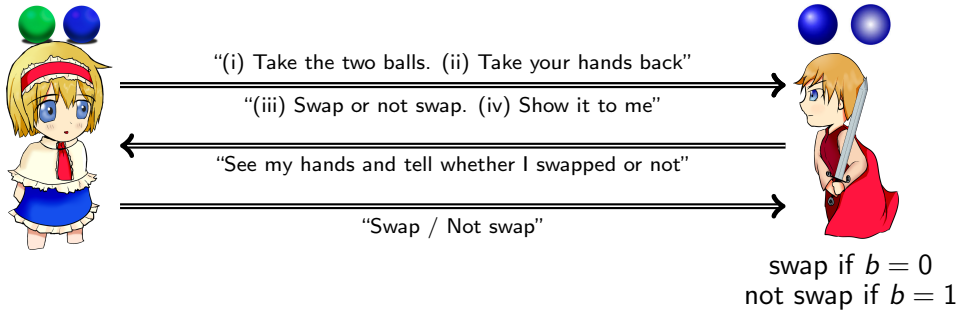
swap if $b = 0$
not swap if $b = 1$

Examples of ZKP : Two Balls and the Color-Blind Friend



- If Alice really knows the balls are distinguishable, then she always wins – (**Complete**)

Examples of ZKP : Two Balls and the Color-Blind Friend



- If Alice really knows the balls are distinguishable, then she always wins – (**Complete**)
- If Alice does not know then she fails with probability 2^{-t} after 't' many repetitions – (**Soundness error**)

Examples of ZKP : Two Balls and the Color-Blind Friend

"(i) Take the two balls. (ii) Take your hands back"

"(iii) Swap or not swap. (iv) Show it to me"

"See my hands and tell whether I swapped or not"

"Swap / Not swap"

swap if $b = 0$
not swap if $b = 1$

After the experiment, Bob does not know which ball is of which color – (**Zero Knowledge**)

Example of ZKP : Discrete Log Problem

Recap of Discrete Log Problem

- G be a multiplicative cyclic group, generated by $\langle g \rangle \in G$.
- Each element $y \in G$ can be uniquely written as $y = g^x$ for some x
- (DL Problem) – Given g and $y \xleftarrow{\$} G$ find x

Example of ZKP : Discrete Log Problem

Recap of Discrete Log Problem

- G be a multiplicative cyclic group, generated by $\langle g \rangle \in G$.
- Each element $y \in G$ can be uniquely written as $y = g^x$ for some x
- (DL Problem) – Given g and $y \xleftarrow{\$} G$ find x
- For cryptographic purpose, we choose $G = Z_p^*$ for prime p of the form $2q + 1$, where q is very large number.

Example of ZKP : Discrete Log Problem

Recap of Discrete Log Problem

- G be a multiplicative cyclic group, generated by $\langle g \rangle \in G$.
- Each element $y \in G$ can be uniquely written as $y = g^x$ for some x
- (DL Problem) – Given g and $y \xleftarrow{\$} G$ find x
- For cryptographic purpose, we choose $G = Z_p^*$ for prime p of the form $2q + 1$, where q is very large number.

Result

- Different sub-exponential algorithms are available, e.g., Pohlig-Hellman, index calculus etc.
- Nevertheless, the problem is hard to solve and the basis for Diffie-Hellman Key exchange algorithm

Example of ZKP : Discrete Log Problem

$$\langle g \rangle = \mathbb{Z}_p^*$$

$$x \in \mathbb{Z}_p^*, y \leftarrow g^x \mod p$$

(g, p, y)

Example of ZKP : Discrete Log Problem

" I want to access my locker"

Alice, (g, p, y)

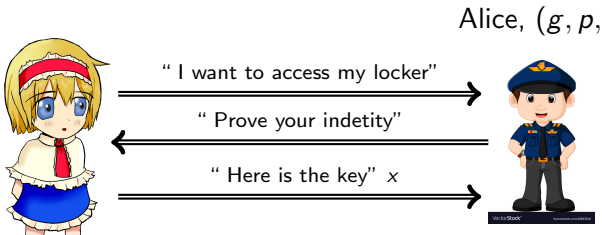
Example of ZKP : Discrete Log Problem

Alice, (g, p, y)

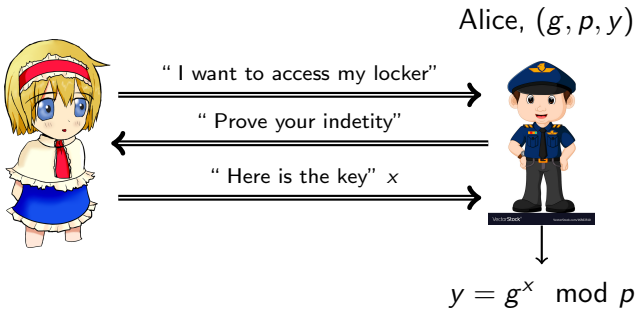
" I want to access my locker"

" Prove your indetity"

Example of ZKP : Discrete Log Problem



Example of ZKP : Discrete Log Problem



Example of ZKP : Discrete Log Problem

Alice, (g, p, y)

" Verified! You can access the locker"

Example of ZKP : Discrete Log Problem

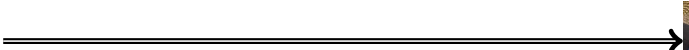
Vulnerability

"Here is the key of Alice, x"

Alice, (g, p, y)

Example of ZKP : Discrete Log Problem

Vulnerability



Example of ZKP : Discrete Log Problem

Vulnerability

Can Alice prove to the guard her secret x in zero knowledge way ?

Example of ZKP : Discrete Log Problem

$r \leftarrow \mathbb{Z}_p^*, h \leftarrow g^r \mod p$

h

Alice, (g, p, y)

Example of ZKP : Discrete Log Problem

$r \leftarrow \mathbb{Z}_p^*, h \leftarrow g^r \mod p$

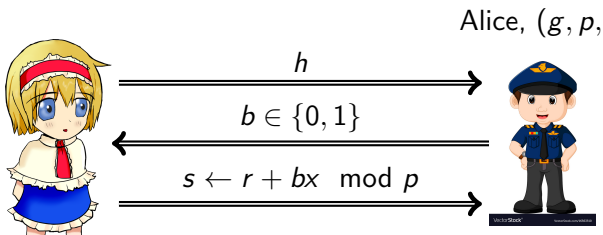
Alice, (g, p, y)

h

$b \in \{0, 1\}$

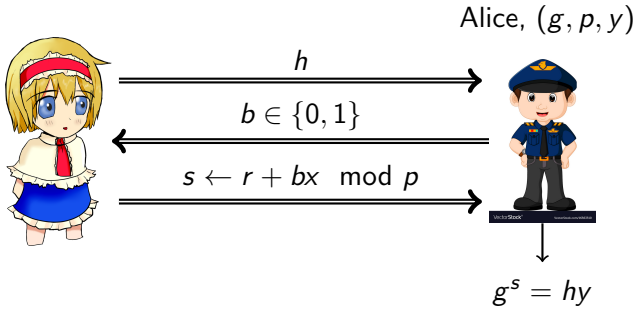
Example of ZKP : Discrete Log Problem

$r \leftarrow \mathbb{Z}_p^*, h \leftarrow g^r \mod p$



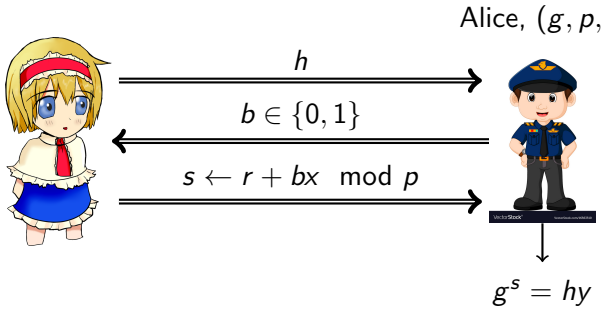
Example of ZKP : Discrete Log Problem

$r \leftarrow \mathbb{Z}_p^*, h \leftarrow g^r \mod p$



Example of ZKP : Discrete Log Problem

$r \leftarrow \mathbb{Z}_p^*, h \leftarrow g^r \mod p$



Repeat the game for 't' times

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?).

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Zero Knowledge: Defining the Simulator

view of the interaction $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) = (h, b, s)$

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Zero Knowledge: Defining the Simulator

view of the interaction $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) = (h, b, s)$

- Pick $b \xleftarrow{\$} \{0, 1\}$

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Zero Knowledge: Defining the Simulator

view of the interaction $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) = (h, b, s)$

- Pick $b \xleftarrow{\$} \{0, 1\}$
- Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Zero Knowledge: Defining the Simulator

view of the interaction $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) = (h, b, s)$

- Pick $b \xleftarrow{\$} \{0, 1\}$
- Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$
- Compute $h = \frac{g^s}{y^b} \bmod p$

Example of ZKP : Discrete Log Problem

Completeness and Soundness

- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Zero Knowledge: Defining the Simulator

view of the interaction $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) = (h, b, s)$

- Pick $b \xleftarrow{\$} \{0, 1\}$
- Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$
- Compute $h = \frac{g^s}{y^b} \bmod p$
- Transcript of the simulator: $M^*(g, p, y) = (h, b, s)$

Example of ZKP : Discrete Log Problem

Completeness and Soundness

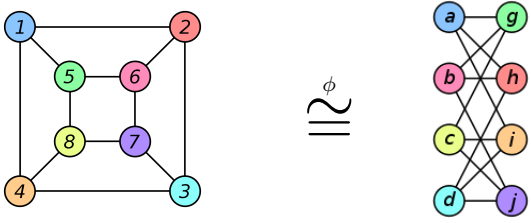
- **Complete** If Alice really knows the secret x, then she will always win the game
- **Soundness** If Alice cheats, then the probability of winning the game in a trial is $1/2$ (how ?). Soundness error: 2^{-t}

Zero Knowledge: Defining the Simulator

view of the interaction $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) = (h, b, s)$

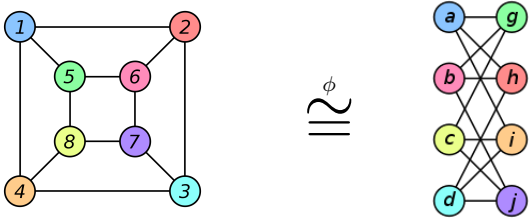
- Pick $b \xleftarrow{\$} \{0, 1\}$
- Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$
- Compute $h = \frac{g^s}{y^b} \bmod p$
- Transcript of the simulator: $M^*(g, p, y) = (h, b, s)$
- Observe that $\text{view}^{\text{Alice}}_{\text{guard}^*}(g, p, y) \cong M^*(g, p, y)$ (**Try at home!**).

Example of ZKP : Graph Isomorphism (GI)



Graph Isomorphism has a zero knowledge proof. In fact GI $\in$ CZK.

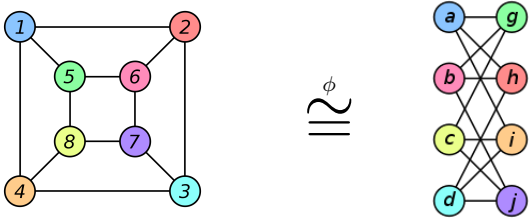
Example of ZKP : Graph Isomorphism (GI)



Graph Isomorphism has a zero knowledge proof. In fact GI $\in$ CZK.

Any NP Language has a Zero Knowledge Proof

Example of ZKP : Graph Isomorphism (GI)



Graph Isomorphism has a zero knowledge proof. In fact GI $\in$ CZK.

Any NP Language has a Zero Knowledge Proof

However, to prove the statement, it requires an another tool called "'commitment scheme"

Applications of ZKP

- Identity Verification: Efficient Replacement of PIN number.

Applications of ZKP

- Identity Verification: Efficient Replacement of PIN number.
- Authentication Systems: Protect User-Passwords across the Internet.

Applications of ZKP

- Identity Verification: Efficient Replacement of PIN number.
- Authentication Systems: Protect User-Passwords across the Internet.
- Cryptocurrencies: Anonymity in Transactions.
  - ZCASH
  - Monero
  - PIVX
  - Zerocoin

Applications of ZKP

- Identity Verification: Efficient Replacement of PIN number.
- Authentication Systems: Protect User-Passwords across the Internet.
- Cryptocurrencies: Anonymity in Transactions.
  - ZCASH
  - Monero
  - PIVX
  - Zerocoin
- eAuction: Bidder's Privacy, Anonymity in the Bidding Amount.

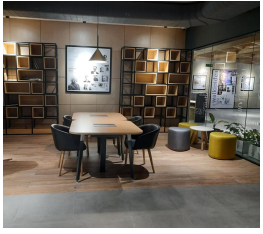
Applications of ZKP

- Identity Verification: Efficient Replacement of PIN number.
- Authentication Systems: Protect User-Passwords across the Internet.
- Cryptocurrencies: Anonymity in Transactions.
  - ZCASH
  - Monero
  - PIVX
  - Zerocoin
- eAuction: Bidder's Privacy, Anonymity in the Bidding Amount.
- eVoting: Prevent a Cheating Voter.

Applications of ZKP

- Identity Verification: Efficient Replacement of PIN number.
- Authentication Systems: Protect User-Passwords across the Internet.
- Cryptocurrencies: Anonymity in Transactions.
  - ZCASH
  - Monero
  - PIVX
  - Zerocoin
- eAuction: Bidder's Privacy, Anonymity in the Bidding Amount.
- eVoting: Prevent a Cheating Voter.

(... and continues)



Thank you for your kind attention !