# Zero Knowledge Proof

## ZKSnarks

Wilson Daubry - 13/05/2020

# Table of content

# Introduction

## Objective of the Zero Knowledge Prove

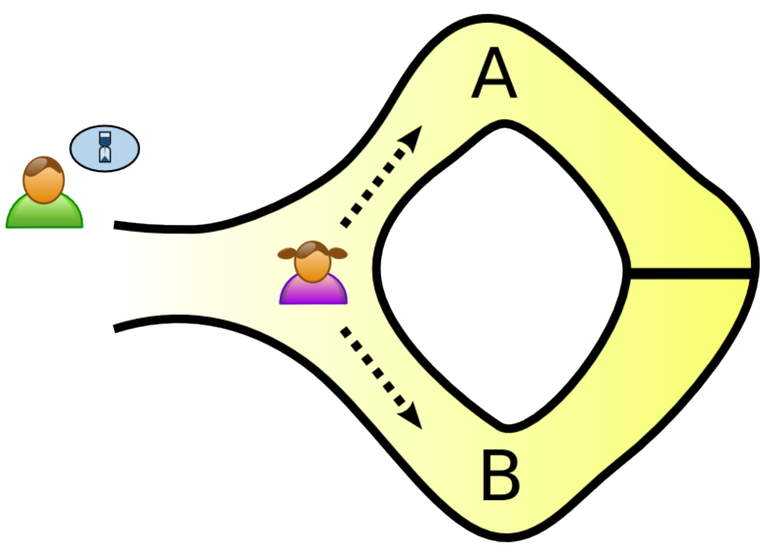
One party proves to another party that he knows an information but without giving any information concerning that information except the fact that he knows the information.

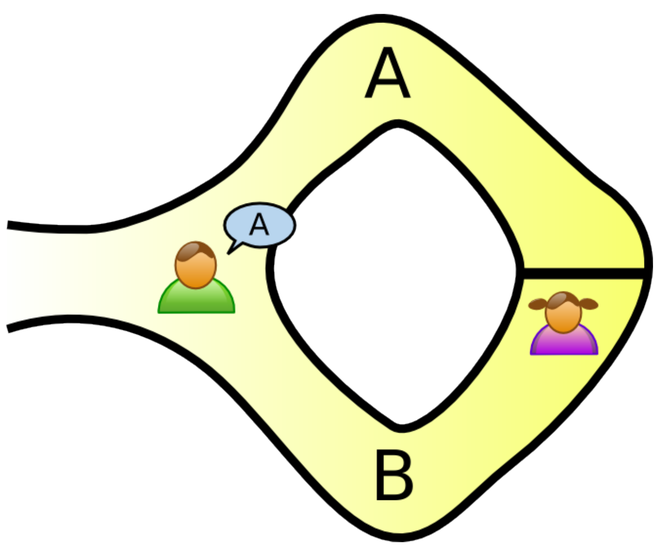
Illustrations :

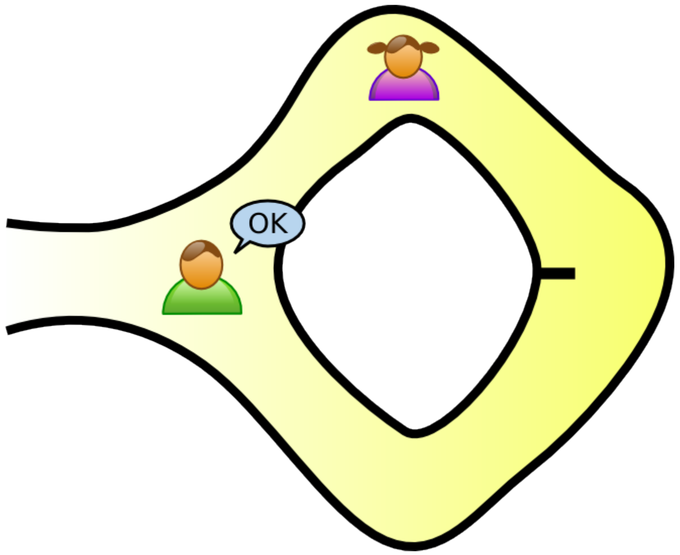
- The Ali Baba cave
- Two balls and the color-blind friend
- Where's Wally ?

# Introduction

## The Ali Baba Cave



# Application of the ZKP

**ZCash**

Cryptocurrency (As Bitcoin)

- Medium of exchange for transaction
- Based on blockchain

Private (Not as Bitcoin)

- Transactions are only known by the actors
- Ledgers content are only known by the owners

# ZKSnarks

**Zero Knowledge Succinct non-interactive arguments of knowledges**

***Succinct*** :

Size of the message small compared to computation.

***Non-interactive*** :

No or few interactions between prover and verifier.

***Arguments*** :

The prover needs to ‘prove’ that he solved the problem.

***Of knowledge*** :

Prover needs to know some information to compute the proof required

# ZKSnarks

## Assumption

Based on an assumption :

- Prover : Big computation power -> Non Polynomial Time
- Verifier : Small computation power -> Polynomial Time

Example :

- Prime number factorization

# ZKSnarks

## NP - Complete

A NP - Complete problem :

- Belongs to the class NP
- Is complete if every NP problem can be reduced to NPC via pol. reduction

Advantages :

- Only one generic ZKSnarks needed
- Every problem is included

# ZKSnarks

## QSP

Chosen problem :

- Quadratic Span Program -> Linear combination of polynomials

Definition : *A QSP over a field F for inputs of length n consists of*

- *Polynomials* $v_0, \ldots, v_m, w_0, \ldots, w_m$ *over F*
- *Polynomial t over F*
- *Injective function* $f : \{(i,j) \mid 1 \leq i \leq n, j \in \{0,1\}\} \to \{1, \ldots, m\}$

$2n < m$

Solution : h, $a_1, \ldots, a_m, b_1, \ldots, b_m$ such that $th = (v_0 + a_1 v_1 + \cdots + a_m v_m)(w_0 + b_1 w_1 + \cdots + b_m w_m)$

# ZKSnarks

## Algorithm of ZKSnarks

Requirement :

- Homomorphic encryption function

Verifier :

- Provides a Common Reference String with in it
  - QSP
  - Encrypted evaluated polynomials
  - Encrypted secret numbers

# ZKSnarks

## Algorithm of ZKSnarks

Prover :

- From the **CRS only**, computes and encrypts :
  - $a_1, \ldots, a_m, b_1, \ldots, b_m$
  - *Polynomial h*

Prover returns those encrypted data along with other datas needed to prove that :

- The solution has been found

# ZCash

## Application of ZKSnarks

Transaction of crypto currency :

- NP problem
- Reduced to a NP - Complete problem (QSP)
- Computes a proof that the transaction is correct
- Every node can be a verifier BUT only of the validity of the transaction