Fun facts about examples of Zero-Knowledge proofs

Ariel Gabizon

Protocol Labs

The deck of cards:

A full deck with red and black cards, face down.

I take out a red three of hearts. How to

convince you I took a red card, without showing which one

Proving color to the color blind:

A red and green ball, otherwise indentical

How to convince a color-blind friend they are different?.

Counting leaves in a tree:

How to prove you can instantly count the number of leaves on a tree, without disclosing the number of leaves?

Visual example: Where's Waldo?

Video: the cave

3-coloring

From interactive to non-interactive

Fiat-Shamir hueristic: simulate challenges of the verifier by hash of messages so far

From interactive to non-interactive

Fiat-Shamir hueristic: simulate challenges of the verifier by hash of messages so far

Homomorphic encryption: Give challenge in advance in homomorphically encoded form (Craig Gentry video)

ZK + bitcoin: Zero-Knowledge contingent payments (by Greg Maxwell)

Chicken and egg problem: I have sudoku puzzle solution, you want to buy it - who goes first?.

ZK + bitcoin: Zero-Knowledge contingent payments (by Greg Maxwell)

Chicken and egg problem: Alice has sudoku puzzle solution, Bob want's to buy it - who goes first?.

ZKCP: Protocol where money and solution change hands at exactly same time.

ZK + bitcoin: Zero-Knowledge contingent payments (by Greg Maxwell)

1. Alice chooses cryptographic key K, sends $h = HASH(K)$.
2. Alice sends encrypted solution $C = E_K(S)$ to Bob; and proves in ZK: "C is encryption of sudoku solution under key who's hash is h.
3. Bob makes bitcoin "hash-locked-transaction" to Alice with h.
4. Alice reveals K to unlock her funds.
5. Bob can now use K to decrypt solution.

More on the mathy side: Schnorr's discrete log protocol

Given g^x, prove you know x without revealing it.

More on the mathy side: Schnorr's discrete log protocol

Given $\mathbf{X} := \mathbf{g}^{x}$, prove you know x without revealing it.

1. Prover chooses random $\mathbf{r}$, sends $\mathbf{R} := \mathbf{g}^{\mathbf{r}}$.
2. Verifier chooses random $\mathbf{c}$
3. Prover sends $\mathbf{u} := x \cdot \mathbf{c} + \mathbf{r}$
4. Verifier checks $\mathbf{X} \cdot \mathbf{R} = \mathbf{g}^{\mathbf{u}}$.