

Zero-knowledge protocols

Problem: **A** wants to identify herself to **B**.

This problem arises for example when

- **A** wants to log on to computer **B**,
- **A** is a customer, **B** an internet bank.

Standard solutions

A identifies herself by her *secret*, e.g., by password or PIN.

In order to avoid that somebody overhearing the communication may learn about **A**'s secret, one may use protocols where

- each password is used only once,
- the secret is not revealed but is just used to solve some task.

Ideal solution

Not even **B** can obtain any relevant information while communicating with **A**, even if **B** deviates from the protocol.

There are such protocols, which are called *interactive proof systems with the zero-knowledge property*.

Zero-knowledge protocols

Definition

Let $G = (V, E)$ be a graph. A k -coloring of G is a mapping

$$g : V \rightarrow \{1, \dots, k\}.$$

A coloring of G is a k -coloring of G for some k . A coloring is *legal* if $g(u)$ and $g(v)$ are distinct for all edges $\{u, v\}$ in E .

A's secret will be a legal k -coloring of a graph G , where it is assumed that it is infeasible to compute a legal k -coloring of G .

Assumption

For the sake of the argument, assume that there is a randomized procedure such that for sufficiently large inputs n and k ,

the procedure yields a graph G with n nodes together with a legal k -coloring of G ,

knowing only G and k , it is infeasible to compute a legal k -coloring of G .

Zero-knowledge protocols

Protocol Coloring

Assumption: **A** knows a legal k -coloring g of G and has access to a random source not known to **B**.

Step 1 (A) Pick uniformly at random a permutation π of $\{1, \dots, k\}$.

Commit secretly to the list of colors $\pi(g(1)), \dots, \pi(g(n))$.

Step 2 (B) Among all edges of G , pick an edge $\{u, v\}$ uniformly at random and send u and v to **A**.

Step 3 (A) Reveal the colors $\pi(g(u))$ and $\pi(g(v))$ to **B**.

Step 4 (B) Accept if the two nodes are colored with distinct colors from $\{1, \dots, k\}$ and reject, otherwise.

In order to diminish the error probability, the Protocol Coloring is iterated m times, where m is the number of edges of G .

Zero-knowledge protocols

The protocol Coloring achieves the following goals

- A** can always verify correctly her identity.
- If at each of the m iterations the colors committed to do not form a legal k -coloring of G , then the probability that **B** accepts is at most $1/2$.
- B** is not able to extract any relevant information while communicating with **A**.

(i): Follows by inspection of the protocol.

(ii): The probability of error is at most $(1 - \frac{1}{m})^m \leq \frac{1}{e} \leq \frac{1}{2}$.

(iii): **B** obtains nothing but mutually independent, uniformly distributed pairs of distinct colors.

In fact, **B** could easily produce his own sequence of pairs of colors that has the same distribution as the sequence evolving during the protocol (= definition of zero-knowledge).

Zero-knowledge protocols

Committing secretly

In the Protocol Coloring, **A** has to commit secretly to a sequence of colors.

If the protocol were executed in real life, this could be done by placing a corresponding number of colored tokens into opaque containers such that **A** cannot change the arrangement afterwards but may reveal the content of any container to **B**.

In electronic form, one would commit to the individual bits of a word describing the sequence of colors.

Committing to a single bit can then be implemented for example by a one-to-one function f that is easy to compute but where for a given function value $f(n)$ it is infeasible to determine whether n has a certain property or not, say, is even or odd.

Here one has to assume that certain functions, e.g., the discrete logarithm, have these properties.