Zero-Knowledge Proofs – final presentation
“The Sudoku Problem”

Pietro Pasotti
1 Live demonstration of the protocol
   Soundness
   Completeness
   Zero-Knowledge
[insert live demonstration of the protocol here]
Live demonstration of the protocol

2 **Soundness**
   Completeness
   Zero-Knowledge
Assume P has the solution. Does V accept with probability $\geq \frac{2}{3}$? Yes, it does: in fact, the probability is 1.
Live demonstration of the protocol

Soundness

3 Completeness

Zero-Knowledge
Suppose P doesn’t have the solution. Then, say it fills in the sudoku blanks randomly. Then, the chances of still picking a correct line/column/square depend heavily on the difficulty of the sudoku. Also, P might use some more refined strategy, such as filling the squares randomly with the numbers 1-9 which are not yet in it. In this case, only lines and columns will contain errors. In the worst-case scenario, where P is maximally smart, only two mistakes can be found in the whole sudoku: two lines/columns/squares which contain a number twice. That is a probability of $\frac{2}{27}$.
However, we know we can have $V$ iterate the procedure many times to take the probability of finding a mistake arbitrarily close to 1. Namely, we will need $V$ to iterate the procedure many times, depending on the size of the sudoku. Remark: this function will be polynomial in the size of the sudoku.

I guess, $|\text{sizeofsudoku}|^2$ could suffice.
It is a good moment to remark that \( V \) is polytime.

- Pick a random line/column/square: polytime.
- Verify the line/column/square contains exactly the numbers \( 1, \ldots, \text{sizeofsudoku} \): polytime.
- Repeat the previous steps a polynomial number of times in \( \text{sizeofsudoku} \): polytime.
- Accept or refuse: still polytime.
- Halt: very polytime.

\( \therefore \) \( V \) is polytime.
Getting closer to the end

Live demonstration of the protocol

Soundness

Completeness

4 Zero-Knowledge
(Perfect) Zero-Knowledge

The simulation routine (S is the simulator):

- Fake Prover fills in the sudoku randomly (or, using some more refined strategy as we saw before).
- Fake Verifier randomly picks a row/column/square: if it’s correct, accept and return 1; if it’s incorrect, S repeats the procedure $n$ times. If, after $n$ iteration, no correct row/column/square has been found, $S$ returns ⊥

To take the probability of finding a correct row/column/square up above $\frac{1}{2}$ it suffices to toy (polynomially) with $n$. I guess $n = \lvert \text{sizeofsudoku} \rvert^2$ again would do. Then, conditional on not returning ⊥, $S$’s distribution is exactly equivalent to the actual one.
The simulator $S$ is polytime

- Fill in randomly the sudoku (there are different strategies, though): polytime.
- Emulate $V$: polytime, because $V$ is polytime.
- Iteration of the procedure polynomially many times: polytime.

$\therefore S$ is polytime.
This is the last slide

That’s all!
Thanks for your attention.