Advanced topic: Zero-Knowledge Proofs

CSCI 3130 Formal Languages and Automata Theory

Siu On CHAN
Fall 2018
Chinese University of Hong Kong
Authentication

- Server knows your password
- They may impersonate you at other websites where you use the same password
Zero-knowledge authentication

Can you convince the server that you know your password, without revealing it?

I know the password

Can you prove it?
Recall that a language $L$ is in NP if there is a polynomial-time verifier $V$ such that $x \in L$ if and only if $V$ accepts $(x, s)$ for some $s$.

Can you prove that $x \in L$?

Verifer $V$ is convinced that $x \in L$, but verifier also knows a lot more.

$s$ is a proof that $x \in L$.
A protocol for non-color-blindness

You want to convince me you are not color-blind

I pull at random either a red ball or a blue ball and show it to you

You say red or blue

We repeat this 10 times

If you got all the answers right
I am convinced you can tell apart red from blue
Interaction and knowledge

What knowledge did I gain from this interaction?

I learned that you can tell apart red from blue

But I also learned the colors of the balls

If I were color-blind

Then I used you to gain some knowledge that I didn’t have
A different protocol

I pull at random either a red ball or a blue ball and show it to you.

We repeat 10 times.

Each time (except the first)
you say “same color as previous” or “different color from previous”.

If you got all the answers right
I am convinced you can tell apart red from blue.

But I did not gain other knowledge!
Suppose I am color-blind but you are not

In the first experiment, I cannot \textit{predict} your answer ahead of time

In the second one, I \textit{know} what you are supposed to say, so I do not gain knowledge when you say it
Graph Coloring

Task: Assign one of 3 colors to the nodes so that every edge has different colors at its endpoints

$$3\text{COL} = \{\langle G \rangle \mid \text{Graph } G \text{ has a valid 3-coloring}\}$$

3\text{COL} is NP-complete

Goldreich–Micali–Wigderson proposed a zero-knowledge protocol for 3\text{COL}
GMW protocol: Choosing a password

password is a random string of colors

\[ \Sigma = \{\text{blue}, \text{red}, \text{yellow}\} \]

e.g. password = ••••••
GMW protocol: Commitment phase

Instead of sending the password to the server, you construct a graph with vertices colored as in password:

1 2 3 4 5 6

-yellow
-red
-blue

Put some (random) edges between vertices of different colors.

Delete the colors of the vertices:

1 2 3 4 5 6

2 5

3 4
Your **real password** is the **coloring**, which you hide from the server.

You give the server a graph $G$ that you know how to color, but the server doesn’t.

Since $3\text{COL}$ is NP-hard, the server shouldn’t be able to figure out your coloring (password) from $G$. 
GMW protocol: Login phase

You randomly permute the colors
You lock each of the colors in an imaginary box
Send the locked boxes to server
Server picks a random edge and asks for keys to the related boxes
You send the two requested keys
The server unlocks two boxes and checks the colors are different
Repeat all of the above steps 1000 times
If colors are always different, login succeeds
Why can’t an impostor log in instead of you?
An impostor does not know how to color the graph
Some edge will be colored improperly
When the server asks to see this edge, impostor will be detected
GMW protocol: Zero-knowledge

Why doesn’t the server learn your password?

When you send the password, the server can only see some locked boxes.

The server then asks you to unlock some boxes.

Colors in the password were shuffled, so server will only see two random colors.
How do you send boxes and keys over the internet?

Commitment scheme!
Other proposed applications

1. Zero-knowledge voting
2. Zero-knowledge nuclear warhead verification