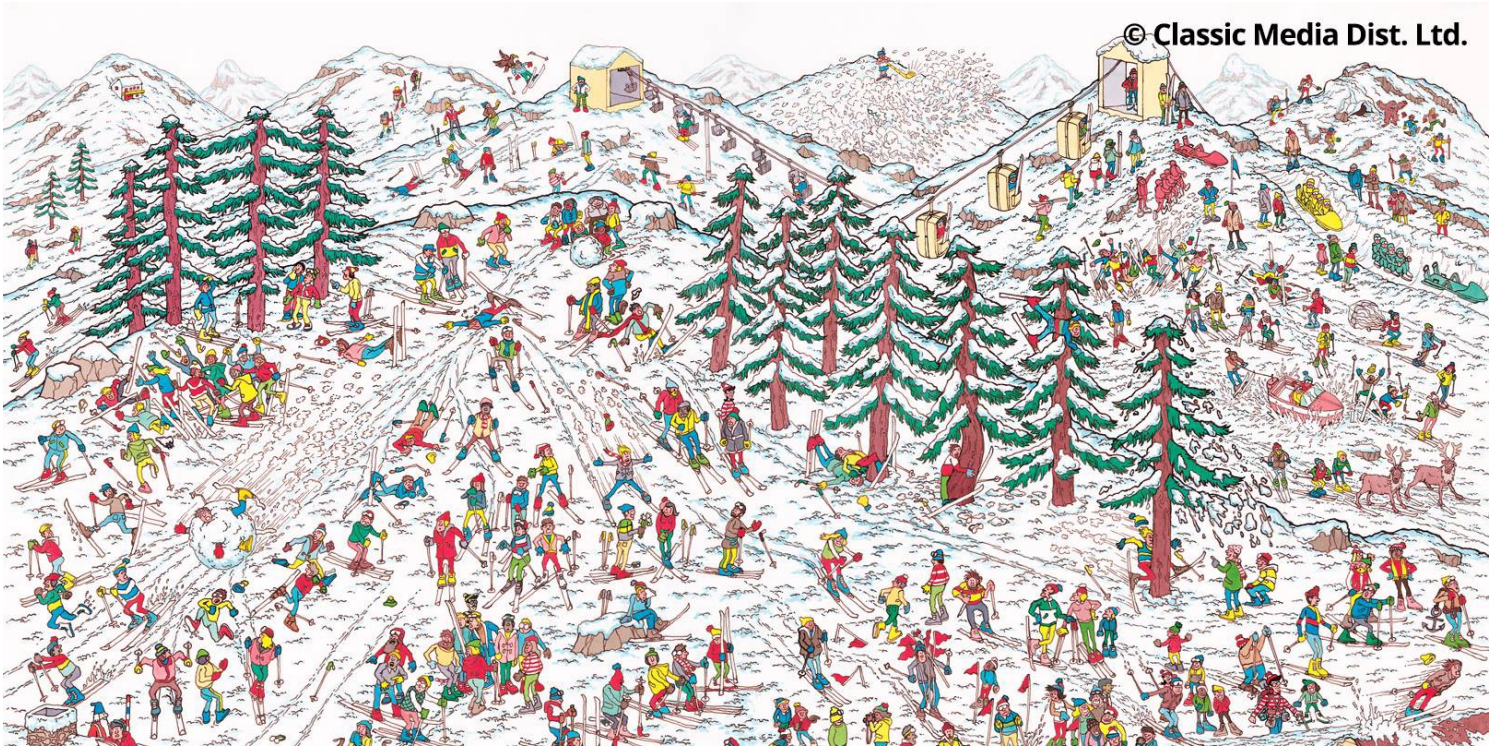


Understanding Zero-knowledge proofs through illustrated examples

This is a somewhat illustrated guide to Zero Knowledge

Nicole Zhu Follow
Apr 8 · 7 min read

I recently listened to this ZKFM podcast, which gave practical examples that clearly explained key concepts in zero knowledge. I felt inspired to transcribe them here.

In cryptography, zero knowledge proofs let **you convince me that you know something,** or have done something, ***without*** **revealing to me what your secret thing was.**

Practically, zero-knowledge is important because it gives you privacy in situations where you'd otherwise have to reveal confidential information. Examples include:

- **Logging into a website**: rather than typing your password into a potentially unsafe website, you can simply send a proof that you "know your password".
- **Authenticating your identity**: rather than giving your mother's maiden name over the phone to a random, bank call center agent, you can simply send a proof (a cryptographic fingerprint), that you are who you say you are.
- **Sending private blockchain transactions**: rather than sending money on Bitcoin, where your financial ledger is public information, you can send a proof that your money is valid (not double spent) without revealing your balances, as popularized by ZCash.

. . .

Example 1: Prove you know where Waldo is, without sharing his location

In this example, Alice and Bob are racing to find Waldo in a popular children's book series, where the point is to spot Waldo in a sea of

shapes that look like him.

The conundrum:

> ***Alice**: I know where Waldo is!*

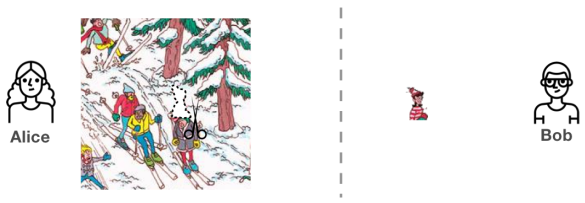
> ***Bob**: Alice, do you know what a liar is?*

> ***Alice**: I can prove to you where he is without revealing his location.*

To defend her integrity, Alice devises two solutions to prove her knowledge.

Proof 1

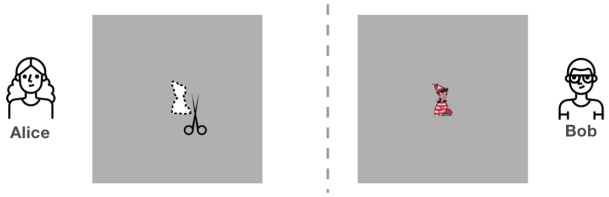
Alice cuts out Waldo from her scene and only shows Bob the Waldo snippet. To ensure that Alice didn't just print out a new picture of Waldo, Bob can watermark the back of Alice's scene page. Or, he can do a thorough cavity search on Alice before Alice enters a secret room to cut the page.



Proof 2

Alice cuts a hole in a very large, opaque sheet of cardboard. She places the cardboard cutout on top of the original scene. In this solution, only Waldo is shown. His coordinates relative to the rest of the scene is still

unknown. Later, Alice can reproduce the scene underneath to prove that she used the original puzzle.



Soundness, Completeness, and Zero-Knowledge

Both solutions fulfill the three important properties of zero-knowledge proof systems: soundness, completeness, and zero-knowledge.

Alice is able to use the same proofs to verify that she has found Waldo many times per game, and across many games. In this sense, her proof systems achieve statistical:

1. **Soundness—everything that is provable is true:** Assuming Alice doesn't know Waldo's locations and presents random pieces of the scene to her proof systems… then, her cardboard holes will display random images without Waldo. *Put simply, Alice's proof systems are truthful and do not let her cheat.*
2. **Completeness—everything that is true has a proof**: As long as Alice finds Waldo, she's able to consistently use her proofs to show Waldo, in each game. *Put simply, Alice's proof systems convince Bob that she found Waldo.*
3. **Zero-Knowledge—only the statement being proven is revealed**: As Alice proves to Bob that she has found Waldo, the only information revealed to Bob is that "Alice has found Waldo". Waldo's location is never revealed. *Put simply, Alice's proof systems prove her victory to Bob, without revealing her knowledge.*

. . .

Example 2: An Interactive, Zero-Knowledge Proof

The following is an example of an **interactive zero-knowledge proof.** In this example, Alice is color blind but Bob is not. She does not believe Bob's claim that he can see colors.

Being the clever cryptographer that she is, Alice procures two balls (one red and one blue) and devises an interactive proof for Bob, using a *challenge and response* mechanism.

> ***Alice**: I am holding a red ball and a blue ball. I will now put my hands behind my back, so you cannot see. During this time, I may or may not exchange the balls between my hands. Afterwards, I will show you the balls again.*

> ***Bob**: Ok, so you want me to tell you if you switched the balls or not.*

> ***Alice**: Correct, only someone who can see color will be able to consistently tell me what I did.*

> ***Alice**: Have I shuffled the balls?*

> ***Bob**: Yes, the balls are in different hands*

> ***Alice**: Have I shuffled the balls?*

> ***Bob**: No, the balls are in the same hand*

In this proof, Alice and Bob will repeat this game many times until Alice is convinced. 20 games later, Alice is convinced that there is only a one in 1,048,576th chance that Bob just got lucky every single guess. She is reasonably convinced that Bob can see color.

This challenge-response example illustrates that zero-knowledge proof systems are only statistical guarantees that something could be true.

. . .

Example 3: A Non-interactive, zero knowledge Proof

The following is an example of a **non-interactive zero-knowledge proof**, which doesn't require the former challenge-response dynamic between Alice and Bob. In this case, Alice can generate all the challenges at once, and then Bob can respond at a later time. Such non-interactive proofs let many parties verify Alice's claims, not just Bob.

The conundrum

Alice wants to prove to Bob and his snooty Sudoku club that she has solved a Sudoku puzzle they have not been able to solve.

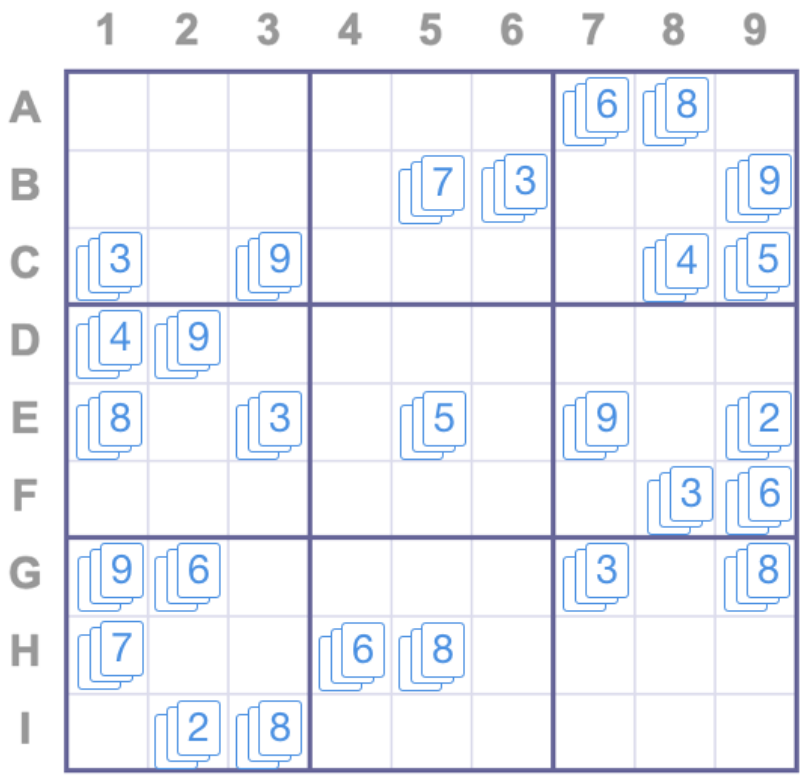
	1	2	3	4	5	6	7	8	9
A							6	8	
B					7	3			9
C	3		9					4	5
D	4	9							
E	8		3		5		9		2
F								3	6
G	9	6					3		8
H	7			6	8				
I		2	8						

Each cell must contain a number (1–9) that is unique to the row, column, and 3x3 grid

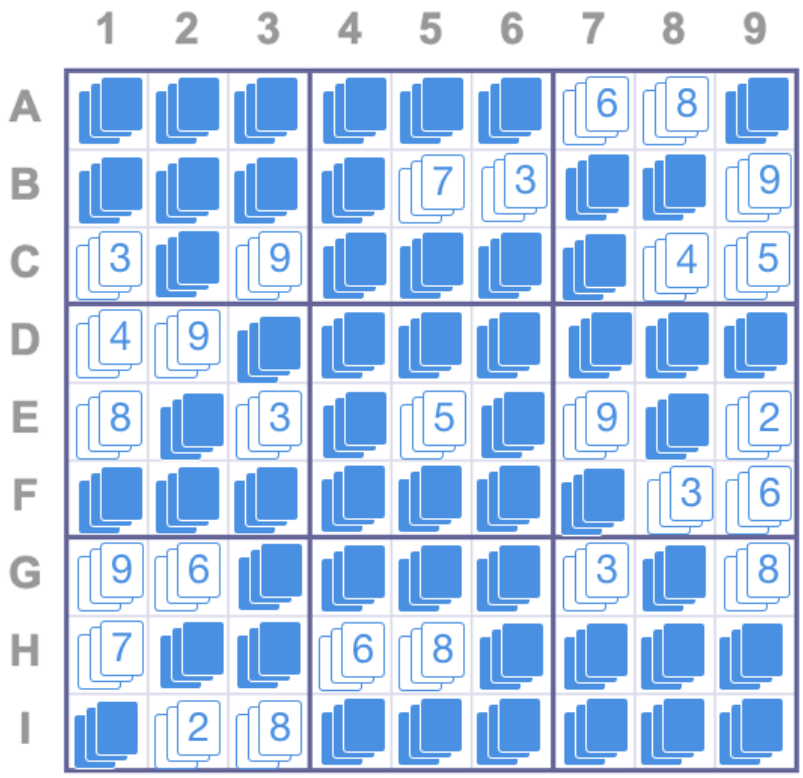
To do so, Alice builds a tamper-proof machine that executes the proof to Bob and friends. Alice's machine follows a specific, *publicly verifiable* protocol with the following logic.

1. First, Alice reproduces the original, unsolved puzzle in the machine. For each cell with an existing value, it automatically lays

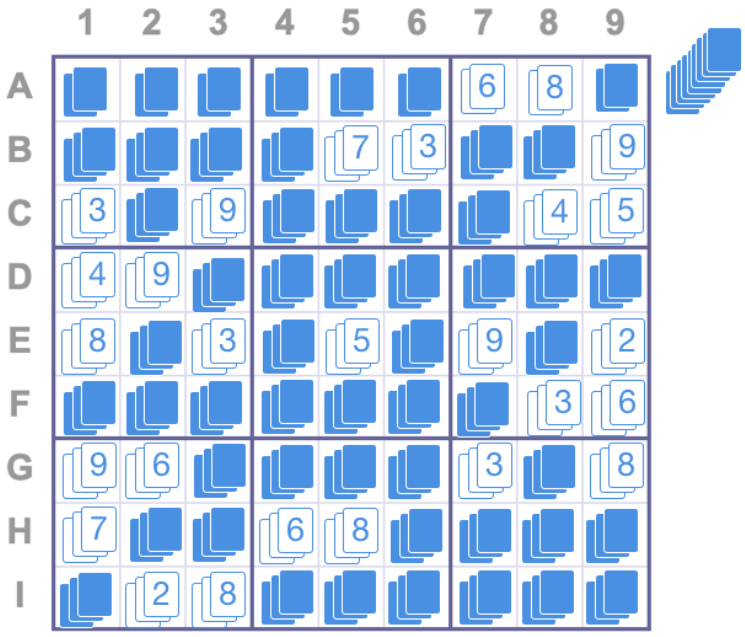
three face-up cards with the corresponding number, e.g. cell C3 has 3 number 9 cards.



2. Next, Alice encodes her solution by having the machine lay her answers face down on the grid. Of course, the machine prevents Bob from simply flipping over the cards in their cells.



3. Bob can now interact with the machine. Starting with each row, Bob randomly chooses one card in each cell, from the top, the middle, or the bottom.



Bob assembles cards for row 1

The machine takes the chosen cards and makes a face down, 9-card-packet for each row.

This action is repeated for each column as well. Finally, the remaining cards are sorted into one packet for each 3x3 grid. In total, the machine makes 27 packets.

4. Then the machine randomly shuffles the cards in each packet, before giving the packets back to Bob.

5. Bob flips the cards over and verifies that each packet contains the numbers 1 through 9 without any numbers missing or duplicated.

The cards are shuffled and no longer have the original order

A few verification rounds later, Bob and friends are convinced that Alice has solved this puzzle.

In this example, the proof is non-interactive. Any one can use the machine (or in reality, code) to verify Alice's claim. Alice doesn't have to be present to be challenged.

There you have it! For a more comprehensive overview checkout the *ZeroKnowledgeFM podcast*.

. . .

Liked what you read?

We're hiring developers to build blockchains & share them with the world! Come join the developer anarchy @ParityTech, and build open source for Web 3.0.

Try it out Substrate, a blockchain builder, here.

Want to do something else? Check out more roles here.

Have someone in mind? Share this post with your network!

. . .

Sources:

- Content sourced from: https://www.zeroknowledge.fm/21
- Waldo images sourced from: https://www.facebook.com/whereswaldo/
- Podcast reference links: Where's Waldo example, Str4d presenting the Billiard Balls example at ZK0x01, Sudoku ZKP example, Eli Ben Sasson presenting STARKs
- Iconography: icons by monkik, asianson.design, Marek Polakovic, from the Noun Project