Zero-knowledge proof

What is Zero-Knowledge proof?

A **zero-knowledge proof** or **zero-knowledge protocol** is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true while the prover avoids conveying any additional information.

ZKP must satisfy three properties:

1. **Completeness**: if the statement is true, an honest verifier will be convinced of this fact by an honest prover.
2. **Soundness**: if the statement is false, no cheating prover can convince an honest verifier that it is true, except with some small probability.
3. **Zero-knowledge**: if the statement is true, no verifier learns anything other than the fact that the statement is true. In other words, just knowing the statement (not the secret) is sufficient to imagine a scenario showing that the prover knows the secret. This is formalized by showing that every verifier has some *simulator* that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" an interaction between an honest prover and the verifier in question.

In basic form, a zero-knowledge proof is made up of three elements: **witness**, **challenge**, and **response**:

- Witness: With zero-knowledge proof, the prover wants to prove knowledge of some hidden information. The secret information is the “witness” to the proof, and the prover's assumed knowledge of the witness establishes a set of questions that can only be answered by a party with knowledge of the information. Thus, the prover starts the proving process by randomly choosing a question, calculating the answer, and sending it to the verifier.
- Challenge: The verifier randomly picks another question from the set and asks the prover to answer it.
- Response: The prover accepts the question, calculates the answer, and returns it to the verifier. The prover’s response allows the verifier to check if the former really has access to the witness. To ensure the prover isn’t

guessing blindly and getting the correct answers by chance, the verifier picks more questions to ask. By repeating this interaction many times, the possibility of the prover faking knowledge of the witness drops significantly until the verifier is satisfied

Types of zero-knowledge proofs:

ZK-SNARKs:
ZK-SNARK is an acronym for Zero-Knowledge Succinct Non-Interactive Argument of Knowledge. The ZK-SNARK protocol has the following qualities:

- **Zero-knowledge:** A verifier can validate the integrity of a statement without knowing anything else about the statement. The only knowledge the verifier has of the statement is whether it is true or false.
- **Succinct:** The zero-knowledge proof is smaller than the witness and can be verified quickly.
- **Non-interactive:** The proof is 'non-interactive' because the prover and verifier only interact once, unlike interactive proofs that require multiple rounds of communication.
- **Argument:** The proof satisfies the 'soundness' requirement, so cheating is extremely unlikely.

ZK-STARKs:
ZK-STARK is an acronym for Zero-Knowledge Scalable Transparent Argument of Knowledge. ZK-STARKs are similar to ZK-SNARKs, except that they are:

- **Scalable:** ZK-STARK is faster than ZK-SNARK at generating and verifying proofs when the size of the witness is larger. With STARK proofs, prover and verification times only slightly increase as the witness grows (SNARK prover and verifier times increase linearly with witness size).
- **Transparent:** ZK-STARK relies on publicly verifiable randomness to generate public parameters for proving and verification instead of a trusted setup. Thus, they are more transparent compared to ZK-SNARKs.

Which problems is it solving?

ZK Proof **facilitates transmitting sensitive information with better privacy and security**. It builds a secure channel for the users to employ their information without revealing it. It circumvents the possibility of data leakage.

Who is using this?

1. Zero-knowledge proofs are also being applied to **anonymizing transactions on public blockchains**. An example is Tornado Cash, a decentralized, non-custodial service that allows users to conduct private transactions on Ethereum. It is using ZK-SNARKs proof.
2. The cryptocurrency Zcash is based on Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK), a type of zero-knowledge cryptographic method.
3. Another example is Zero-Knowledge Scalable Transparent Argument of Knowledge (zk-STARK), which is used in the Ethereum blockchain and provides privacy and scalability.

Where can we use this?

1. Finance
2. Online voting
3. Authentication
4. Personal identification, such as a driver's license, Social Security card or passport.
5. Proof of income, such as W-2s, paystubs or filed tax returns.
6. Employer's information, including the company name, your manager's name and the phone number.
7. Proof of residence, such as a utility bill with your name and address or a lease agreement.

Challenges?

- Not 100% secure, some probability exists for unsecure.
- Computational Intensity.

Opportunities:

- Because of this web3 market will pick a very good demand.
- Blockchain and crypto also pick very good demand.