DIGITAL IDENTITY, PRIVACY, AND ZERO-KNOWLEDGE PROOFS (ZK-SNARKS)

Adam Luciano Follow
May 25, 2018 · 8 min read

Our personal data (date of birth, social security number, education credentials, work histories, etc.) helps create a picture of who we are, our identity. In the wake of Facebook's data being mined [1] by Cambridge Analytica and Equifax's massive data breach [2] (a company that holds our most personal data, including web usage statistics,

financial information as well as medical data), I think there is much to be improved in how our data is handled by any such organization. The concept of a self-sovereign identity [3], where organizations will no longer own or even keep our data is right around the corner. We as the individuals that worked to establish all these important components of our identity will own and manage it. The idea of self-sovereign identity will be discussed in more detail in a future blog post, however there are still ways to fight for our right to privacy even if some of our data has been exposed to hackers and marketers, especially around data that creates our identity.

Protection of data related to the identity of individuals is vitally important and will continuously increase in importance as we share more of our data electronically. Data can primarily be compromised in two ways: when at rest (example: on your hard drive, or in your wallet), or when in motion (example: sending data via an email or pulling out your driver's license in a crowded restaurant). I would argue that data in motion (sharing with a third party) is riskier than data at rest, as data in motion is available to prying eyes (or snooping algorithms). For example, if my driver's license is in my wallet, there is much less of a chance that my date of birth will be seen by an unsavory character as opposed to when I remove it from my wallet.

The data that we constantly create about ourselves is up for grabs, however recent innovations in cryptography and blockchains enable a system to be created that can help protect our data and identity, even from organizations that we interact with. How do you really know if you can trust the organizations that you interact with? Two obvious weak links in the chain of trust are the very people we interact with at said organizations, and their unsecured IT systems. These weak links are universal across organizations that we rely on for various products and services, such as our electric company, or internet provider. These weak links could lead to our data being compromised, as there are multiple individuals, organizations and state actors that seek and pay for such breached data. Enter a method called Zero-Knowledge Proofs[4].

Zero Knowledge Proofs enable a way to share validated data with a third party without actually sharing the data itself. This is a fascinating concept, as intuitively it takes a little time to wrap one's mind around the fact that you can share sensitive information without actually sharing sensitive information. Zero-knowledge proofs work in the following way: If I would like to prove something to someone (let's say Person A), I

generally would need to have a piece of evidence that backs up my claim to Person A, who is trying to verify my claim. Zero-knowledge proofs create an irrefutable way to prove that I have evidence to back up my claim, by sharing a proof instead of the evidence with a verifier that can be used to validate my claim. Using a zero-knowledge proof, I do not actually need to share any information/data, but a cryptographic proof (that does not leak any data) to prove my claim to Person A. Zero-Knowledge Proofs effectively provide a riskless way to share data (assuming we trust the process of setup that creates the proof, more on that later).

There are countless ways that we can use Zero-Knowledge Proofs in the future to protect our data. One way that has not been widely discussed in realistic terms and that solves two big problems is calling into call centers. Suppose Alice needs to call her bank to initiate a transfer from one of her bank accounts to another bank account. Wouldn't it be great if when Alice called the bank, instead of going through a process of answering questions about herself to verify her identity (account number, social security number, address, mother's maiden name), Alice could send a cryptographic proof from her mobile phone to the bank that automatically authenticates her identity? That would save time, and increase privacy… Let's look at how the call would go in this example after Alice authenticates her identity on the phone:

Example: Bob responds to Alice after Alice authenticates her identity via the current process. (Average time to address the issue Alice called about — 30–60 seconds)

Bob (Call Representative)

"Good Morning, my name is Bob. Can you please provide me with your full name?"

Alice

"Alice Smith"

Bob (Call Representative)

“Thank you. Can you please provide me with your current address?”

Alice

“12345 Main Street Springfield MA 99999”

Bob (Call Representative)

“Thank you so much for the information. Can you please provide me with the last 4 digits of your social security number?”

Alice

“Sure, 6789”

Bob (Call Representative)

“Thank you so much. Can you please provide me with the amount of your last transaction on your checking account?”

Alice

“I don’t remember”

Bob (Call Representative)

“Thank you so much for your patience. I am really sorry, but we really need this information in order to proceed.”

Alice

“let me check…. $123”

Bob (Call Representative)

“That’s great, thank you so much for your patience. I am really sorry about the delay. We are now able to proceed. How can I help you Miss Smith?”

Alice

“Can you transfer $100 from account 123 to account 456?”

Bob (Call Representative)

“Sure, I will start the process immediately. Is there anything else that I can help you with?”

Alice

“No, that is it. Thank you!”

Example: Bob responds to Alice after Alice authenticates her identity via her mobile phone using a Zero-Knowledge Proof Approach for smart contract enabled blockchains. (Average time to address the issue Alice called about — 5–10 seconds.)

Bob (Call Representative)

“Good Morning Alice, my name is Bob. How can I help you?”

Alice

“Can you transfer $100 from account 123 to account 456?”

Bob (Call Representative)

“Sure, I will start the process immediately. Is there anything else that I can help you with?”

Alice

"No, that is it. Thank you!"

As you can see in example #2 the identity process supported by our Zero-Knowledge Proof is a lot shorter than if Alice had to prove her identity to Bob, and then ask for the bank transfer. Also, Alice did not risk sharing any information about herself with Bob the call center representative. In example #1, Alice doesn't know if Bob wrote it down, or if the bank's systems are secure, etc.

What is a zero-knowledge proof exactly? A Zero-Knowledge Proof enables one party (Alice the prover) to prove to another party (Bob the verifier, the call center agent) that Alice can prove that Bob is speaking with Alice (Alice is looking to prove her identity). All this occurs without revealing any of Alice's private and sensitive information while proving that Alice is in fact Alice.

Typically, the way Bob would believe that Alice is Alice, is to ask Alice questions about herself to prove to Bob that Alice is in fact Alice (ex: What is your address, social security number, mother's maiden name, etc.). However, this relies on sharing information, which can be costly as well as become a reputation risk if the data is leaked through the bank's computer system and shared with unauthorized third parties.

There is a form of a zero-knowledge proof called a zk-SNARK (Zero-Knowledge Succinct Non-Interactive ARgument of Knowledge) that can be used to satisfy the condition of proving Alice is Alice with 100% certainty and without revealing any more information than the cryptographic proof generated from the zk-SNARK. A key element in the zk-SNARK is the non-interactive element: there only needs to be one set of data that is shared with Alice's Bank, hence there is no interaction between Bob and Alice where they exchange sensitive and private information to prove Alice's identity. Remember: the cryptographic proof shared cannot be linked to the data that was put into the creation of the proof. This means that if Alice provides her social security number to the Zero-Knowledge Proof program, she can be 100% confident the verifier (Alice's Bank) will not be able to derive her social security number from the proof Alice provides. Utilization of zk-SNARKs start us on a path where we can retain ownership of our data by simply not sharing sensitive data any longer, thus significantly increasing privacy, and significantly

reducing risk for organizations. In a world of Zero-Knowledge Proof usage, I expect to see Equifax, and the type of leaks it allowed to be a thing of the past.

In my next post, I will dive deeper into Zero-Knowledge Proofs by further clarifying what they need to be in order to be considered a true zk-SNARK, as well as further expand the call center example to provide a more realistic way of how it could work in organizations today.

This post can also be found on Cynapse Blockchain Consulting's website: https://cynapseblockchain.com/2018/09/07/digital-identity-privacy-and-zero-knowledge-proofs-zk-snarks/

In addition, if organizations have questions around ZK tech or tokenomics, please feel free to review more how I help organizations at https://cynapseblockchain.com/

Part 2 of this blog series can be found here.

Blockchain Zksnark Digital Identity Zero Knowledge