Zero-Knowledge Proofs and Blockchain

Alexander C. Mueller, PhD

June 26, 2017

Who is the speaker?

30 second resume:

- an ancient metro Saint Louis townie
- grew up in University City
- University City High School 2003
- B.A. Washington University 2007 (econ and math)
- Ph.D. University of Michigan 2013 (math)
- a few years of data science
- founded data privacy company Capnion

Agenda

Zero-Knowledge Proofs

- What is a zero-knowledge proof?
- Zero-Knowledge Examples

Blockchain Privacy

- Double Spending
- Storing Hashes

SNARKs: What and Why?

- Unpacking SNARK

Application in Zcash

- Privacy via Zero-Knowledge
- “The Ceremony” and “Moon Math”

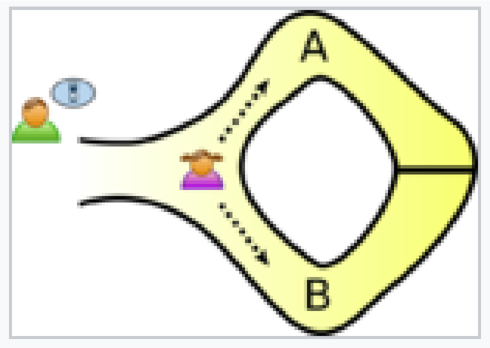
What is a zero knowledge proof?

A **zero-knowledge proof** is a method by which a *prover*, traditionally called Peggy, proves to a *verifier*, traditionally called Victor, that

- Peggy possess a particular piece of information, and...
- Peggy does not reveal that information to Victor.

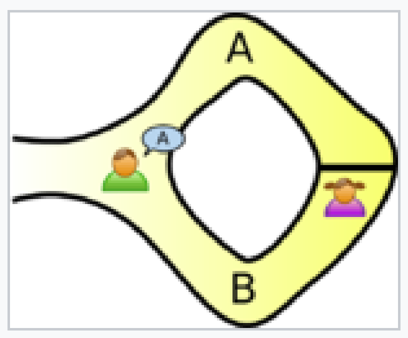
Example Many digital signature algorithms fit our definition well: Peggy proves to Victor that she holds the private key corresponding to a particular public key but Victor does not receive any information about the private key itself.

Classic Cave Example 1



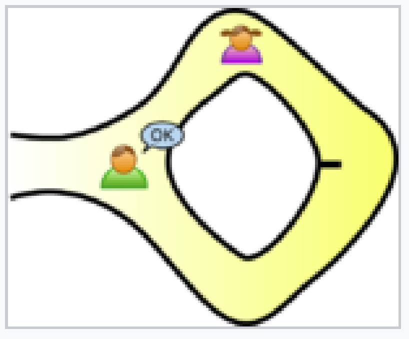
Peggy wishes to prove to Victor that she can open the magic door in the back of a donut-shaped cave, yet she wants to keep her method for opening the door a secret from Victor.

Classic Cave Example 2



After Peggy has entered the cave by one path or the other, Victor shouts into the cave whether Peggy should return by Path A or Path B.

Classic Cave Example 3



If Peggy can consistently return by the path Victor requests, she has demonstrated that she has a method for passing through the magic door, yet Victor has not seen Peggy open the door.

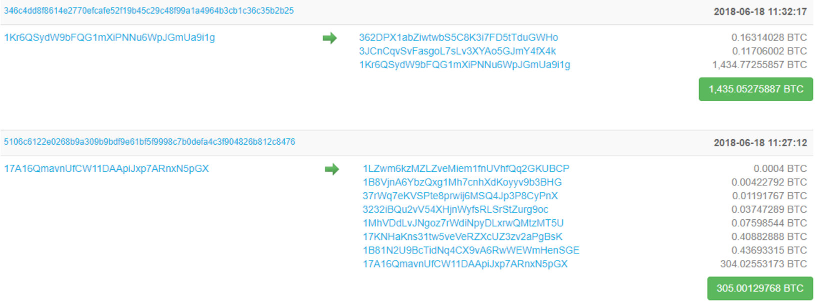
Blockchain Transparency 1

https://www.coindesk.com/194993-btc-transaction-147m-mystery-and-speculation/

194,993 BTC transaction worth $147 Million sparks mystery and speculation

Blockchain Transparency 2

https://blockchain.info/largest-recent-transactions

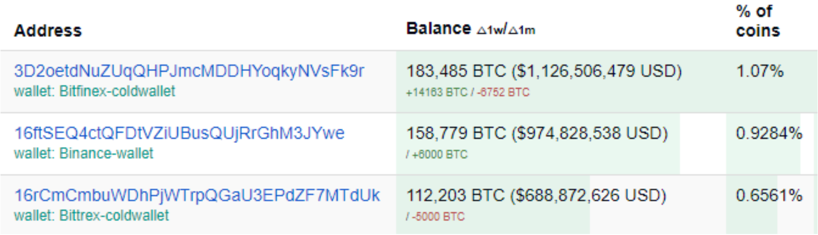


346c4dd8f8614e2770efcafe52f19b45c29c48f99a1a4964b3cb1c36c35b2b25			2018-06-18 11:32:17
1Kr6QSydW9bFQG1mXiPNNu6WpJGmUa9i1g	➡	362DPX1abZiwtwbS5C8K3i7FD5tTduGWHo	0.16314028 BTC
		3JCnCqvSvFasgoL7sLv3XYAo5GJmY4fX4k	0.11706002 BTC
		1Kr6QSydW9bFQG1mXiPNNu6WpJGmUa9i1g	1,434.77255857 BTC
			1,435.05275887 BTC

5106c6122e0268b9a309b9bdf9e61bf5f9998c7b0defa4c3f904826b812c8476			2018-06-18 11:27:12
17A16QmavnUfCW11DAApiJxp7ARnxN5pGX	➡	1LZwm6kzMZLZveMiem1fnUVhfQq2GKUBCP	0.0004 BTC
		1B8VjnA6YbzQxg1Mh7cnhXdKoyyv9b3BHG	0.00422792 BTC
		37rWq7eKVSPte8prwij6MSQ4Jp3P8CyPnX	0.01191767 BTC
		3232iBQu2vV54XHjnWyfsRLSrStZurg9oc	0.03747289 BTC
		1MhVDdLvJNgoz7rWdiNpyDLxrwQMtzMT5U	0.07598544 BTC
		17KNHaKns31tw5veVeRZXcUZ3zv2aPgBsK	0.40882888 BTC
		1B81N2U9BcTidNq4CX9vA6RwWEWmHenSGE	0.43693315 BTC
		17A16QmavnUfCW11DAApiJxp7ARnxN5pGX	304.02553173 BTC
			305.00129768 BTC

Blockchain Transparency 3

https://bitinfocharts.com/top-100-richest-bitcoin-addresses.html



Address	Balance △1w/△1m	% of coins
3D2oetdNuZUqQHPJmcMDDHYoqkyNVsFk9r wallet: Bitfinex-coldwallet	183,485 BTC ($1,126,506,479 USD) +14163 BTC / -6752 BTC	1.07%
16ftSEQ4ctQFDtVZiUBusQUjRrGhM3JYwe wallet: Binance-wallet	158,779 BTC ($974,828,538 USD) / +6000 BTC	0.9284%
16rCmCmbuWDhPjWTrpQGaU3EPdZF7MTdUk wallet: Bittrex-coldwallet	112,203 BTC ($688,872,626 USD) / -5000 BTC	0.6561%

Informal Definitions

The following informal definitions should be enough to get beginners through the next few slides.

- A blockchain is a database. All changes to its state are time-stamped, visible to all users, and once committed can never be retracted.
- A hash function takes an arbitrary document as input and outputs a string. Two documents (almost) never produce the same output string, and it is extremely difficult to start with an output string and then find an input document that produces it.

Zero-Knowledge: Blockchain Non-Examples

Much of the traditional blockchain architecture uses what one might call total-knowledge proofs.

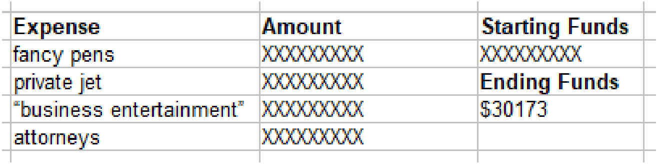
- Bitcoin makes all transaction amounts publicly visible. A payee can verify a payer has enough coins by examining their history, and the history of their coins, all the way back to the miners. This solves the double spending problem.
- Blockchain notaries provide proof-of-existence by publishing a hash of the document on the blockchain. The document's creator can prove its past existence by sharing the document and pointing to the prior existence of the hash.

Both these examples require considerable disclosure of information.

Application to Digital Notaries

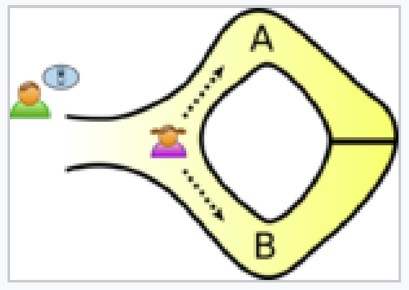
Consider the extra privacy provided by ZK to digital notaries

- TK: here is a document that produces the hash previously stored on the blockchain.
- ZK: I possess a document that produces the hash previously stored on the blockchain, and this document has the following additional properties.



Expense	Amount	Starting Funds
fancy pens	XXXXXXXXX	XXXXXXXXX
private jet	XXXXXXXXX	**Ending Funds**
"business entertainment"	XXXXXXXXX	$30173
attorneys	XXXXXXXXX	

Cave Zero-Knowledge for Blockchain?



It sounds like a cool idea to get rid of our total-knowledge proofs and replace them with zero-knowledge proofs, but...

Why doesn't the cave example generalize well to blockchain applications?

ZK-SNARK

ZK-SNARK abbreviates **Z**ero-**K**nowledge **S**uccinct **N**on-interactive **AR**gument of **K**nowledge. The two notable new properties are...

- **non-interactive** Peggy sends information to Victor one time, and never requires any response from him.
- **succinct** Peggy doesn't need to send a large amount of information to Victor.

How does the cave example fail to be SNARK?
Why is the succinct property important for blockchain?

Why SNARK is important?

The payer and payee need to leave some record of their transaction on blockchain for others to, and any static record is necessarily **non-interactive**.

It is expensive to store data primarily on a blockchain, so it is important that any record left on the blockchain is **succinct** (or more simply put, not too big).

These two extra properties allow blockchain to serve as the substrate for zero-knowledge proofs. **Zcash** is a cryptocurrency that uses this technology to eliminate the need for public transaction amounts.

Proving What? Classic Example

I can store proof that I solved a particular Sudoku without sharing the solution of that Sudoku.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Proving What? Hidden Ledgers

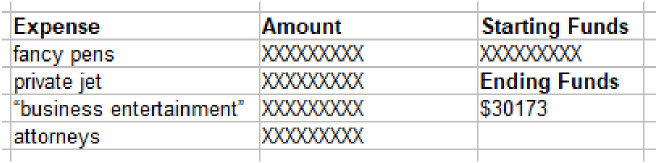
You can use a ZK-SNARK to show you posess a solution any NP-complete problem (a huge class of problems) without giving the data of the solution. Some important, non-exclusive examples...

- I have data that produces a given hash, and...
- My data fits into a certain schema, and...
- Certain algebraic relations hold.

In our old spreadsheet example, this is the assertion you can do a simple, blind audit on a ledger as it was when its hash was published to the blockchain.

ZK-SNARK Blind Audit Revisited

I possess a document that produces the hash previously stored on the blockchain, and this document has the following additional properties.



Expense	Amount	Starting Funds
fancy pens	XXXXXXXXX	XXXXXXXXX
private jet	XXXXXXXXX	Ending Funds
"business entertainment"	XXXXXXXXX	$30173
attorneys	XXXXXXXXX	

I have only added some restrictions to the basic problem of computing a document that produces a given hash. A ZK-SNARK can be used to prove the existence of a solution to this type of problem without disclosing the data of the solution.

Application in Zcash

Zcash uses ZK-SNARKS to solve the double-spending problem. The flow is approximately...

- Any transaction (including those from mining) leaves an encrypted “audit receipt” of cryptographic data.
- Prior to a transaction, the payee uses their old audit receipts to construct a ZK-SNARK that states that they have records of doing honest accounting and that this accounting shows they have enough money.
- After the transaction, the new audit receipt reflects that the payer spent coins and the payee received them, but this information is only visible to the parties affected.

The blockchain holds just enough information to demonstrate honest accounting, but not enough to reveal details.

Some History and Big Ideas

ZK-SNARKs are unusually new and sophisticated...

- They are the product of an unusual burst of academic creativity over the past 10 years, and...
- They overlap a good deal with the related, exciting field of homomorphic encryption.
- By way of comparison, the constituent cryptographic technologies in a blockchain are decades old.

A couple landmark papers and people...

- (Gentry 2009) “A Fully Homomorphic Encryption Scheme”
- (Gennaro, Gentry, Parno, Raykova 2013) “Quadratic Span Programs and Succinct NIZKs without PCPs”

Intuition for how they work: Polynomial factorization

Cryptography usually works via a computation that is much more difficult in one direction than the other. For d numbers $\alpha_1, ..., \alpha_d$ you can define a polynomial...

$$P(z) := \prod_{i=1}^{d} (z - \alpha_i)$$

If we define a polynomial this way, we know that

$$P(\alpha) = 0 \iff \alpha = \alpha_i \text{ for some } i$$

It is easy to verify claims that some number is a root, it is easy to construct a polynomial with given roots, but it is hard to factor a polynomial.

Steps of ZK-SNARK

Steps and Ingredients in the ZK-SNARK:

- A: A common reference string
- B: Your problem specified in a certain format
- $(A \wedge B) \Rightarrow$ Compute a complicated polynomial

The polynomial is constructed so that a solution to the problem can easily be used to compute a root but roots are difficult to reverse-engineer into solutions.

The polynomial is very difficult to factor without knowledge of the solutions to the problem. To prove I have a solution, I share the roots I computed using my solution.

“The Ceremony” and Common Reference String

The ZK-SNARKs used in Zcash require a common reference string *uniform across users of the currency* and a presumption that the string was created in a particular way. Too much knowledge of the creation of the string grants vast power, so an exotic creation ceremony was devised...

- Six people contributed secret data used to create the CRS.
- The creation procedure was constructed so all six must collude to compromise the CRS.

The stakes are high, as collusion among the six would allow them to mint coins, etc.