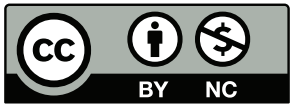
Security Aspects of Blockchain



Blockchain

- What is the blockchain?
- Why is the blockchain?
- What is it good for?
- What are the security risks of it? To it?
- Disclaimer: I am not a blockchain fan—it has its uses, but not always what its proponents say
- It was a magnificent technical achievement

What is “the” Blockchain?

- A *distributed ledger*
- “Distributed”: not run by any single party
- “Ledger”: Records transaction history, rather than just instantaneous state
- Not limited to money—but that was its first use and the goal of the inventor(s)

Who Invented the Blockchain—and Why?

- The original Bitcoin paper was posted pseudonyously by “Satoshi Nakamoto”
- There have been many “revelations” and a few claims, but none are generally accepted (or, to me, acceptable)
- Why was it simply posted, and not submitted to an academic venue?
- It was, I would say, revolutionary enough to have won “best paper” at more or less any security or privacy conference
- Conclusion: the inventor(s) wanted a real-world effect, not academic laurels

Who Were the Cypherpunks?

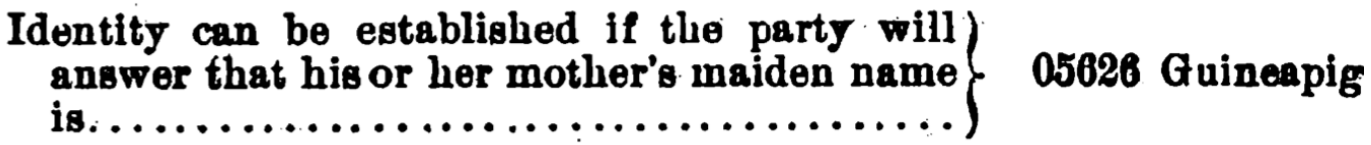
- A 1990s group of cryptography enthusiasts—some (but not all) were very knowledgeable
- Philosophically, the group was libertarian-anarchist
- Basic goal: use cryptography to make governments go away
- (No, I don't know why this would have worked, or why they felt it would be good if it did happen.)
- Tactic: strong cryptography for privacy, especially financial privacy

Was Satoshi Nakamoto a Cypherpunk?

- In my opinion, almost certainly
- At the very least, Nakamoto's design (appeared to have) satisified their design goals for a cryptocurrency—and these goals were often stated on the mailing list
- And: Nakamoto was not an academic, or the paper would have been formally published

Prior Digital Currencies

- Electronic transfers of money go back to the telegraph age



- Credit card numbers over the Internet are also obvious
- The trick was *privacy-preserving* digital transactions

Chaum, Fiat, Naor

- Privacy-preserving digital cash dates to 1988 (and some say 1982)
- It and all subsequent digital cash schemes (until Bitcoin) required a trusted party: effectively, a bank
- The hard part of digital cash: preventing double-spending of digital "coins"

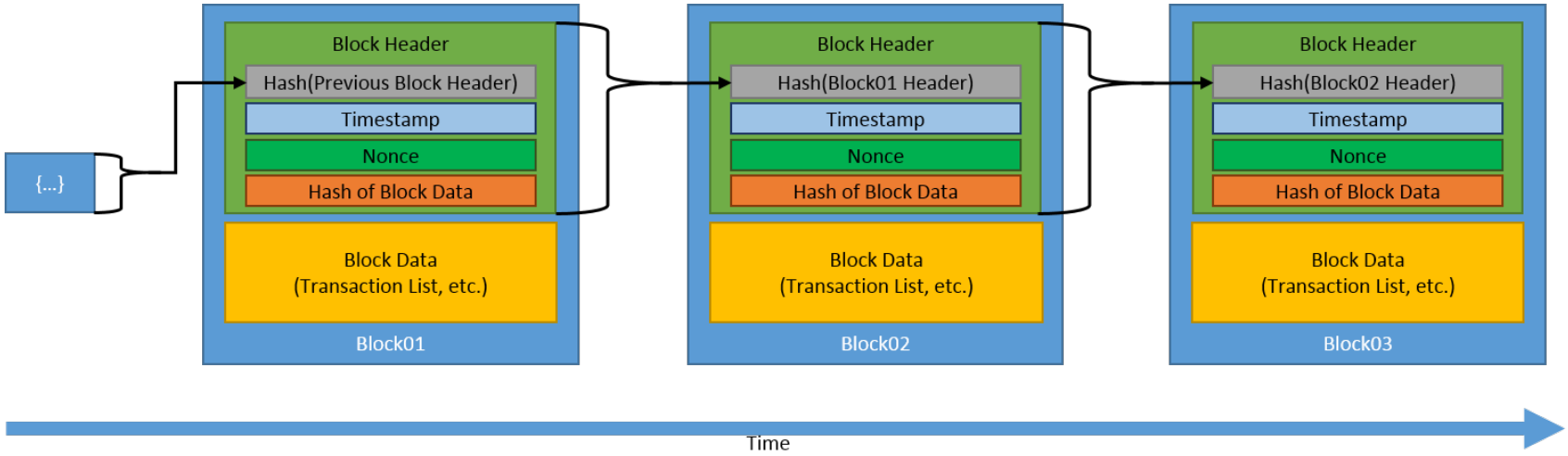
Enter the Blockchain

- Nakamoto's paper was the first digital cash scheme that did not require a bank
- Instead, it required rough agreement among a majority of nodes
- "The system is secure as long as honest nodes collectively control more CPU power than any cooperative group of attacker nodes."

Nakamoto's Goals

- Non-reversible transactions—Nakamoto believed that reversibility would require financial institutions and merchants to collect more private data
- Permit small, casual payments
- No trusted third party—use peer-to-peer verification

The Basic Blockchain



(Source: https://doi.org/10.6028/NIST.IR.8202)

Basic Structure

- Each block (via a hash) is an input to the next block
- This links blocks together
- Crucial innovation: proof of work
- The hash of a block must meet certain constraints

Proof of Work

- Every block includes a nonce
- The hash of a block—*including the nonce*—has to have N leading zeroes
- No better way (believed) possible than to iterate over nonces until the hash comes out right
- ☞ To add a **block** to the **chain**, it's necessary to find the proper nonce; this takes *lots* of CPU (or custom hardware) cycles
- However, it's quick to verify that a hash is correct
- N is determined by the block creation rate: the more blocks are being created, the more zeroes are necessary, so more computation is needed

Growing the Blockchain

- Every new block is shared with all other full Bitcoin nodes
- When two new blockchains arrive at a node, the longest wins
- If there's a tie, the first to arrive wins

Implementing Currency on the Blockchain

- Every party has a private/public key pair—a Bitcoin “address” is the (hash of) the public key
- Bitcoin blocks have a particular format: a set of input (coin) addresses and a set of output (payee) addresses, both with values, all digitally signed by the owner,
- The input values have to equal the output values (not quite true—stay tuned)
- You can direct an output back to yourself, as change
- Multiple transactions, from multiple people—currently, about 2000—form a single block that is added to the blockchain

Scripts

- Each payment can also include a small script written in a special scripting language
- Scripts can be used to implement conditions on the payment: multiple signatures, effective date of the payment, etc.
- This language is *not* Turing-complete
- There are no loops, so termination is guaranteed

What is a Coin?

- To Nakamoto, a “coin” is a “chain of digital signatures”
- In other words, every coin must appear on the blockchain
- A coin, then, is something that you were paid, traceable back over all coins that you have ever received

Mining

- Why should anyone bother calculating these hashes? Two parts to the answer. . .
- First: if the input values total more than the output values, the difference is a payment to the “miner”
- Second: there’s a payment for succesfully adding a block to the chain, currently 25 Bitcoins
- This is how Bitcoins are created: payments to the “miners” who do these hash calculations

☞ Miners can preferentially add more profitable transactions to the blockchain first

Double Spending

- The trick with any digital cash system is preventing double-spending
- That is, if I have a Bitcoin, what's to stop me from paying it to two different people?
- With Bitcoin, all transactions are recorded on the blockchain
- That is: the validity of any payment is determined by finding a payment to its Bitcoin address of that amount
- But no coin can be spent twice, because spending it is also on the blockchain

Storing Bitcoins

- Most people stored their bitcoins on “exchanges”
- They’d log in to their exchange account to make transactions
- Today, there are “hardware wallets”—(supposedly) secure devices to protect private keys

Nakamoto's Goals

Small, casual transactions No—transactions take too long and are too expensive. It takes too long to find the right nonce for today's values of N

Privacy Bitcoin is not anonymous, it's *pseudonymous*. Nakamoto recognized that but the threat is more serious than was realized then

Decentralized As Nakamoto realized, Bitcoin was only secure "as long as honest nodes collectively control more CPU power than any cooperating grou of attacker nodes". That's not always true these days.

But... The distributed ledger was a genuine innovation

Essential Properties of the Blockchain

- The blockchain is a *ledger*, a record of transactions
- It is sometimes described as “immutable”; as we shall see, that is not true
- *Anything* can be recorded on the blockchain, not just Bitcoin transactions
- However, there needs to be some incentive for the miners to calculate hashes—the payments to the miners

Smart Contracts

- Suppose the scripting language were more powerful
- You could impose powerful contract fulfillment conditions
- These are called “smart contracts”—they’re self-executing, with no need for courts to enforce them
- Ethereum is based on this idea

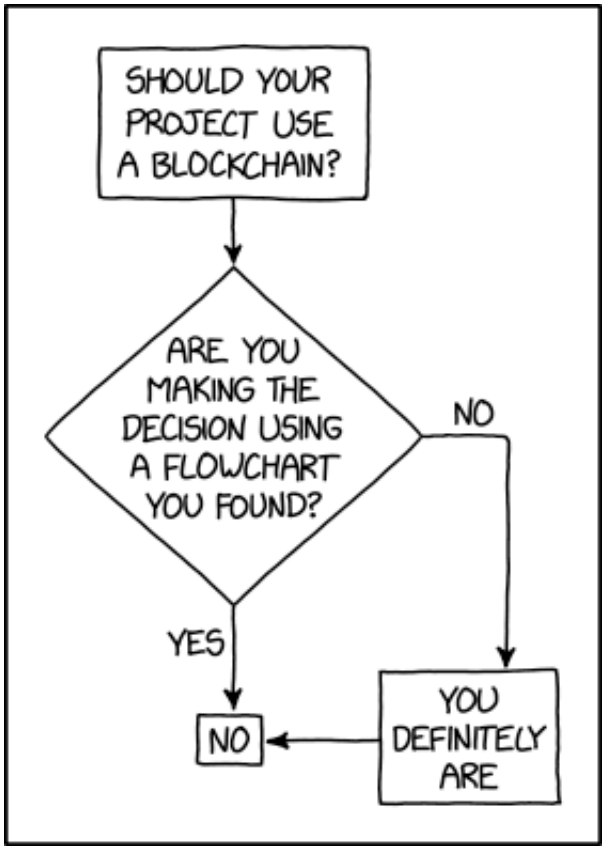
CryptoKitties!

- Virtual cats on the blockchain, using Ethereum
- Each “cat” has “genetic material”, for color, stripes, etc.
- You can breed cats; they’ll all look different, depending on what genes they inherit
- For a while, they were very popular—someone once paid $140,000 for one
- No, I’m not joking

"Permissioned" (Closed) Blockchains

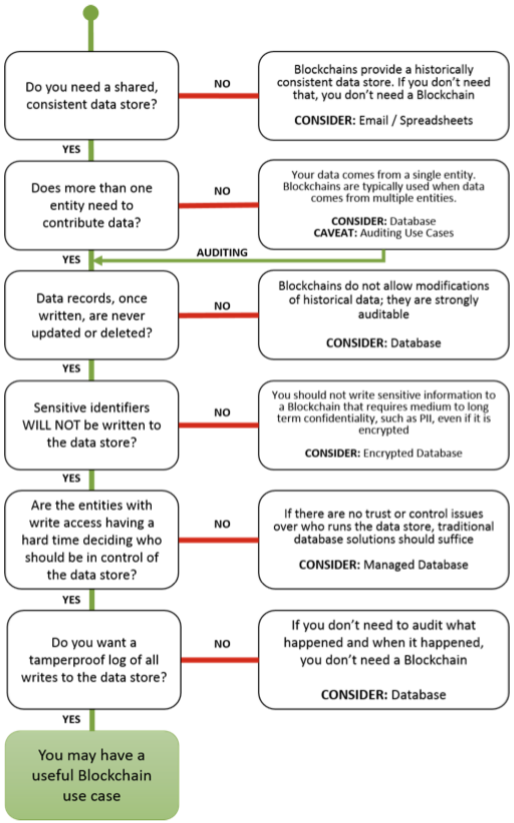
- Not all blockchains are available to the public
- Read access can be separate from write access
- Some require authorization to use; others are inside intranets
- As we shall see, this solves many of the security and scaling problems
- But—is blockchain really necessary?

Should You Use the Blockchain?



(https://xkcd.com/2267/)

A More Nuanced View



(Source: `https://doi.org/10.6028/NIST.IR.8202`)

Essential Blockchain Properties

- There is no one trusted party
- Multiple writers, multiple readers
- Near-immutability
- You can afford the expense

If all of those properties hold, a blockchain might be suitable

Security Analysis of Blockchain

- What security problems does it solve?
- What are the security risks?
- Do alternatives have similar risks?
- In other words, is there an incremental risk to blockchain?
- If so, is it outweighed by blockchain's advantages?

Attacker Goals

- Theft of money
- Changing supposedly immutable data
- Denial of service
- Privacy breach

The Math

- The cryptographic math behind Bitcoin and the blockchain appears to be correct
- Note: “appears to be”—cryptographic protocols are notoriously hard to get right
- The protocols have been examined carefully—but don’t be surprised if a flaw is found
- (Today’s blockchain—the vital historical continuity that says which coins are valid—is vulnerable to quantum computers. . .)
- That’s not the biggest problem. . .

Buggy Code

- As usual, bad code beats good crypto
- There have been many, many bugs affecting the actual security of Bitcoin
- Ethereum is even worse

Random Numbers

- Cryptography is heavily dependent on random numbers
- For example: Bitcoin private keys are supposed to be random
- One Android app got that badly wrong
- In fact, it was arguably negligently wrong

Non-Random Randomness

- The app did *everything* wrong
- It downloaded randomness from a website (`random.org`)
- At the time, that website used HTTP, not HTTPS, allowing others to eavesdrop on the random numbers used to generate a key pair
- It didn't switch to the HTTPS website when told to
- It sometimes didn't mix in local randomness

☞ Result: several people had the same private and public keys, and hence the same Bitcoin address. . .

Smart Contract Flaws

- Ethereum smart contracts are written in code
- Code, of course, can be buggy
- But the code is on the blockchain, and hence is immutable
- Oops. . .

The DAO Hack

- DAO: Decentralized autonomous organization
- Scheme to fund Ethereum projects; naturally, it's implemented as a smart contract on the blockchain
- Lots of people invested
- There was a bug in the basic DAO code. . .
- It allowed an attacker to steal Ethereum from the DAO
- And the buggy code was unfixable because it was on the blockchain

Counter-Thieves

- The good guys wrote their own theft code—to move the DAO's funds to another address, to be restored later
- Was this legal?
- "You literally have cyber ninjas warring on the blockchain"

Hard Forks

- How can the underlying code be fixed?
- Remember that the blockchain is not *actually* immutable: “The system is secure as long as honest nodes collectively control more CPU power than any cooperative group of attacker nodes.”
- If a majority of the mining power agrees, the block chain can be altered
- The idea was put to a vote; 97% of Ethereum users agreed with the concept: fix the bug *and* retrieve the stolen money
- But some purists didn’t want to go along—so they “forked” the blockchain to create Ethereum Classic
- Both versions survived!

Forking for Evil

- Nakamoto noted that a majority of miners needed to be honest
- On Ethereum Classic—the “no hard fork” branch—some attackers obtained control of a majority of the mining capacity
- They used this to delete transactions from the blockchain, which let them double-spend

Centralizing Decentralization

- With the blockchain, power belongs to the miners
- Mining today requires specialized hardware to compute hashes rapidly
- It also requires cheap electricity—Bitcoin consumes more electricity than Switzerland
- It uses more electricty than all of the world's banks—but processes only $\frac{1}{5000}$ as many transactions
- $\frac{2}{3}$ of the world's mining capacity is in China—far more centralized than Nakamoto anticipated

More Smart Contract Bugs!

- One thing you can do with Ethereum is create “multi-sig” coins, ones that require multiple signatures to spend
- This relies on a code library; that code, of course, is on the blockchain. . .
- There was a bug in some code that permitted an attacker to take over ownership of that library—and to disable it
- “I accidentally killed it”
- Result: $280M in Ethereum coins became unuspendable, because they relied on a library that was permanently—thank you, blockchain—disabled
- Ironically, the buggy code was introduced to fix another multi-sig bug that let $31M be stolen—and only $31M was stolen because of more “good” thieves

Hacking: Clients

- Bitcoins are spent by signing transactions with a private key
- How do you protect those keys?
- Sometimes, poorly
- Back in 2011, someone lost 25,000 Bitcoins when his home PC was hacked

Hacking: Servers

- Most people stored their wallets—and private keys—on Bitcoin exchange servers
- Those can get hacked, too
- Hundreds of millions of dollars, perhaps billions, have been stolen that way
- Plus: some exchanges have been accused of being fraudulent to start with
- (Many other legal issues I won't go into)

Hardware Wallets

- Today, you can store your private key in a hardware wallet
- In essence, these are personal HSMs
- Are they secure? What is their attack surface? Are they hackable? History suggests they're not likely to be that trustworthy—but at least they can be unplugged when not in use

Privacy Issues

- Bitcoin is, as noted, pseudonymous, not anonymous
- If a person is ever linked to a Bitcoin address, all transactions to and from that address are linked to that person
- (That can happen if a computer is seized by law enforcement—and given how often Bitcoin is used for illegal activities, that's a real risk for some people. . .)
- Bitcoin transactions form a graph—and it's often possible to analyze these graphs
- Information leakage, e.g., IP addresses, can permit some reidentification
- Active attacks can find more people

Providing Privacy

- First attempt: tumblers
- Tumblers: people put Bitcoins in; withdraw almost the same amount (the difference pays for the service)
- This breaks the linkage (on the blockchain) between Bitcoins in and Bitcoins out
- Harder to get right than it seems

ZCash

- Zcash is a cryptocurrency—it's not compatible with Bitcoin—the provides provably strong privacy guarantees
- It relies on sophisticated cryptography, including zero-knowledge proofs

How Do We Scale Blockchain?

- Right now, the transaction rate for Bitcoin is far too low to meet Nakamoto's vision
- As a corollary, transactions are very expensive
- It is clear that the original architecture *cannot* scale enough—no one did the engineering calculations
- There are (complex) proposals for scalable blockchains
- It is likely that at least some cryptocurrencies will move in that direction

Proposed Uses of the Blockchain

Currency Enough said!

Public Records Maintain (mostly) immutable logs of, e.g., property records. Esepcially useful in places where government isn't that honest

Supply Chain Track the provenance of everthing that went into a product

Privacy Preferences Let individuals record their preferences, for all webs sites, data brokers, etc., to see

Elections An awful idea, for many reasons

Some of these will be public blockchains; others will be permissioned.

Blockchain

- The original blockchain idea was quite innovative
- However, it was incapable of scaling far enough to meet its goals; it was never engineered
- There are newer variants that seem able to solve those problems—but what is the use case?
- Most problems can be solved more easily with older techniques
- Permissioned blockchains have some uses—but careful analysis is required
- And blockchain's cryptography does not make it immune to security problems

So Who Was Satoshi Nakamoto?

- Some people speculate that it was Hal Finney, a cypherpunk who was the first Bitcoin user after Nakamoto
- Finney denied it
- Nakamoto created the first two blocks on the blockchain, and hence the first Bitcoins (and could do mining on an ordinary PC)
- Ability to spend coins from the first block—the "Genesis Block"—would be proof of possession of the private key used to sign it
- None of the claimants have shown that they have this ability