RSAConference2020
San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: ACB-T12

Distributed Trust: Is “blockchain” the best approach?

Radia Perlman

Fellow
Dell Technologies
Radia.Perlman@dell.com

Charlie Kaufman

Security Architect
Dell EMC
Charlie.Kaufman@dell.com

We constantly hear

- Centralized = BAD
- Distributed = GOOD
- Distributed = Blockchain!!
- What exactly does “distributed” mean?
- What exactly does “blockchain” mean?

RSAConference2020

Beware the Hype!!!

Blockchain

- Started as the technology behind Bitcoin
- People made money on Bitcoin
  - The more hype, the more money gets poured in
- Startups leverage the hype to claim their product has something to do with “blockchain”
  - And the money rolls in
- After hearing so much hype, natural to assume “blockchain” must be important

Hype

- Articles about how “blockchain” is:
  - Biggest advancement in technology since the Internet
  - “Being considered for” all sorts of problems
  - “Even the US Military is looking at blockchain technology to secure nuclear weapons"

So what is blockchain?

- Very difficult, with all the hype, to actually find out how “blockchain” works…most of what is written just says that it’s a ‘gamechanger for everything’ and how much various companies are investing
- And there are so many variants of ‘blockchain’, it’s hard to separate a ‘blockchain’ technology from a ‘non-blockchain’ technology
- We won’t cover all the details, or all the variants; just enough to understand what the properties generally are
- We’ll focus on what is “distributed trust”, and compare what’s accomplished with “blockchain” vs alternative approaches

RSAConference2020

Blockchain: The Beginning

Where did the term “blockchain” come from?

Bitcoin

- Paper in 2008, by Satoshi Nakamoto (presumably a pseudonym)
- The design powering Bitcoin was called ‘blockchain’

Bitcoin Design Goals

- Don’t trust any known organizations (banks, governments, etc.)
- Prevent governments from doing things like:
  - Shutting the system down
  - Enforcing tax collection
  - Following a money trail
  - Preventing transfer of money to terrorist organizations
  - Inhibit collection of ransom money
- Whether or not those are good goals
  - It does lead to an interesting design
  - With a lot of misinformation about its characteristics
- What, if anything, does it provide that couldn’t have been done before?

Note: To save time, we'll simplify a bit

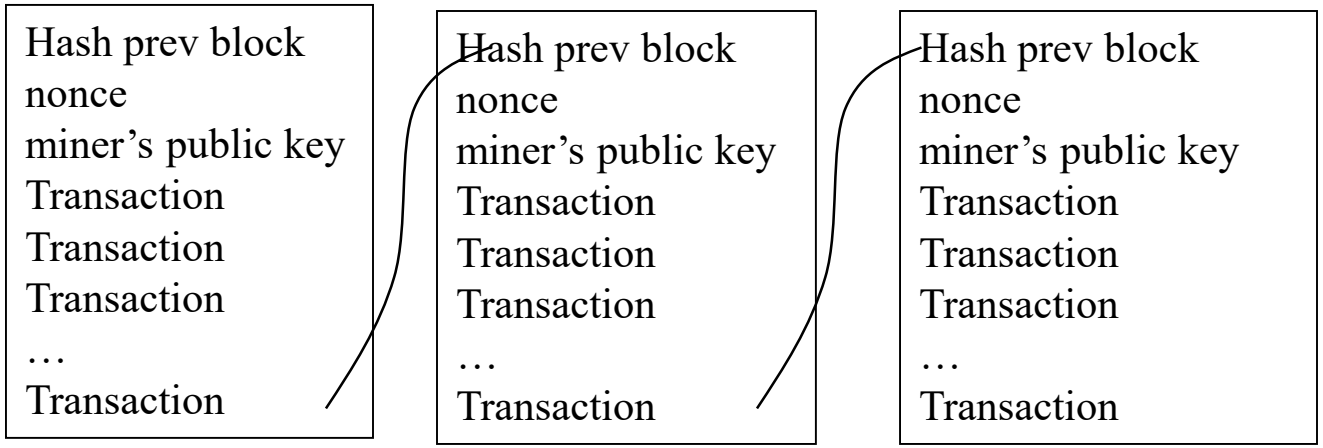
So what is “blockchain”?

- A bunch of anonymous independent nodes (which we’ll call “miners” or “ledger maintainers”) collaborate to create a “ledger”
- FYI..In Bitcoin, items on the ledger are records of the form: public key A sending some amount of Bitcoin to public key B (signed by A)

So what's a "ledger"?

- Is a "ledger" a revolutionary innovation?
- It's just an append-only log
- Most systems have audit logs, but usually discard log data older than some age (e.g., a month, a year)
- Distributed Database technology predates blockchain, and is much more general than blockchain
  - Allows data to be organized conveniently
  - Allows data that is obsolete to get deleted

Format of blockchain



Hash is "preimage resistant", so infeasible to find other data with the same hash

So what's not new

- The concept of keeping a log
- Having data structured as a linked list of blocks
  - Merkle trees are much older than "blockchain", and much more general…a blockchain is a Merkle tree with no branching

What is new?

- Integrity check without authorized entities
- The ability to dynamically adjust the difficulty of finding the next block
- We'll explain…

Cryptographic Hashes

- A good hash is like a random number
  - As if, for every input, a random number were generated
- Probability that 1st bit = 0 for random input is 50%
- Probability that top 10 bits = 0 for random input is $1/2^{10}$
- Bitcoin's blockchain adjusts the hash difficulty (how many leading zeroes) so that a new block added about every 10 minutes

Miners/ledger maintainers

- If you're lucky enough to find the next block, you are rewarded with Bitcoins
- Build on the longest valid chain you've seen (verify that all the transactions in the chain are valid)
- To create a block
  - take new valid transactions, and a random number, and compute the hash
  - the hash has to have a certain # of leading 0's (be smaller than some value)
  - If the hash is too big, change the random # and try again
- The difficulty (number of leading 0's) is adjusted so that on average it takes about 10 minutes for one of the miners in the community to find a block

Miners/ledger maintainers

- Finding the next block is a little like winning a lottery
  - you get rewarded if you win
    - a fixed reward for the block
    - plus any transaction fees – optional “tip” to miner
  - the more you invest (compute in blockchain, money in lottery), the more likely you’ll win
- Currently the hash needs to have 71 leading 0’s
- 2^{71} is a very large number

RSAConference2020

Immutable

"Immutable"

- Can't change the data without being detected
- Why is (Bitcoin's) blockchain hard to forge?
- Assumption: It took so much compute to find blocks, that nobody else could possibly create a different chain
  - If an alternative, longer chain is introduced, the old chain will be forgotten
  - So if attackers could compute an alternate valid-looking chain, they can undo history, and double-spend
- Assumes the blockchain mining community has more compute than any nation-state, etc., could possibly put together

In contrast…with traditional cryptography

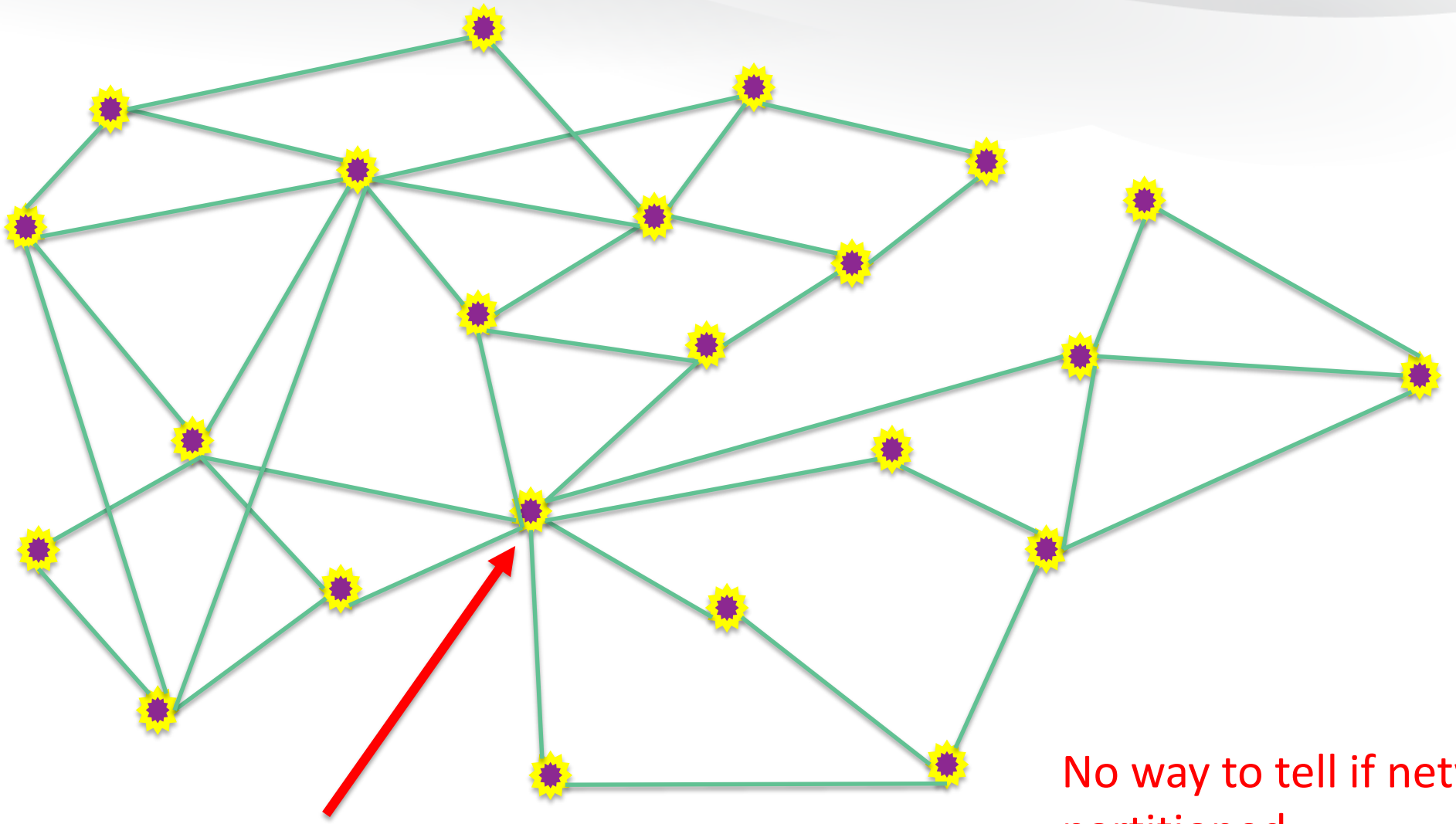
- Public keys…authorized nodes know a secret (private key)
- Cryptography makes a huge gap between the compute necessary to create a signature vs forge
- For example, with 2048 bit RSA
  - To sign: about 6 milliseconds on a typical CPU
  - To forge: the entire compute power of the Bitcoin mining community for the next million years
- In contrast: Blockchain equally expensive to compute as to forge

To summarize

- Traditional cryptography
  - Known, authorized parties have secrets (private keys)
  - More secure (exponentially more difficult to forge)
  - And incredibly less compute-intensive
- Bitcoin's mining community does have a huge amount of compute power
  - Currently using 7.7 Gigawatts
  - Largest US nuclear power plant (Palo Verde) generates 4 GW
- But with zillions of other cryptocurrencies being created, they can't all have that much compute power…

Communication

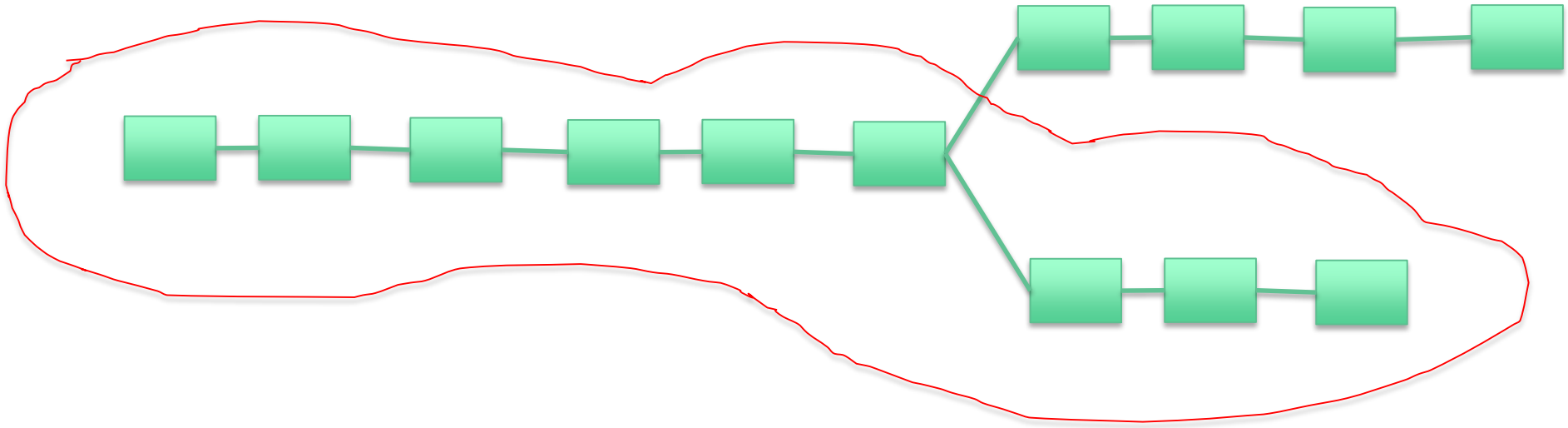
- The Internet doesn't have a mechanism for "send this to all Bitcoin miners"
- So instead, need to configure miners to have "links" to several other miners
- With lots of redundancy, multiplies the numbers of messages
- Without lots of redundancy, a few miners going down can cause partitions



If this node goes down, network is partitioned

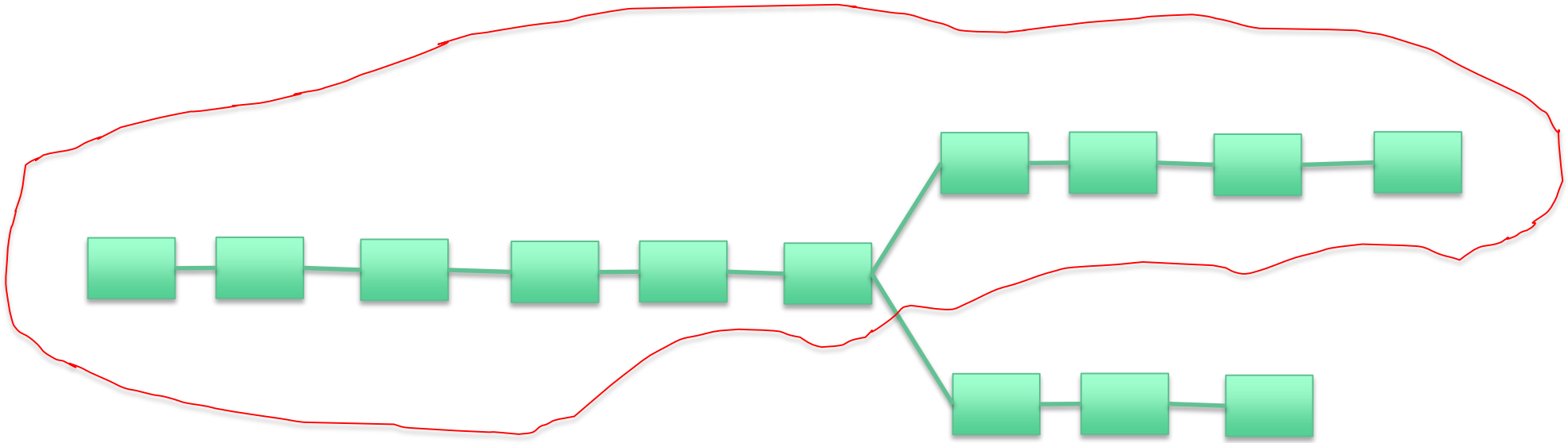
No way to tell if network is partitioned

Forks



Valid chain

Forks



Also valid chain: Longer, so erases the shorter chain

All transactions in losing chain, all Bitcoins mined, no longer exist

What causes a fork?

- An attacker with more compute power introduces a longer chain
- A network partition
- Variable message delays, semi-simultaneous finding “next block”
  - Bitcoin “rule of thumb” is wait for 6 extra blocks before you can feel safe that your transaction will stay
    - That’s an hour!
    - But there’s actually no amount of time that guarantees your transaction will stay
- Disagreement about what a “valid transaction” is
  - This actually occurred (March 2013)
  - “Someone” had to make a decision about which fork lost
- Disagreement about changes (e.g., block size)
  - Caused a permanent fork into two currencies (Bitcoin and Bitcoin cash)

Using Bitcoin

- Suppose someone pays you in Bitcoin
- You get a message saying some public key P pays your public key what they received in some previous transaction, that had hash H1
- Will you be paid?
  - Have to find, somewhere in the ledger, a transaction with hash H1
  - P has to have been the recipient of that transaction
  - Have to search the entire ledger after transaction H1 to make sure P hasn't already paid that to someone else
  - Then you have to make sure the transaction is "solidly" in the ledger (or else P can pay it to someone else)
- This is expensive (e.g., keeping the entire ledger)
- So you ask a "full node" whether to trust the transaction
- It answers "yes" or "no" and you trust it

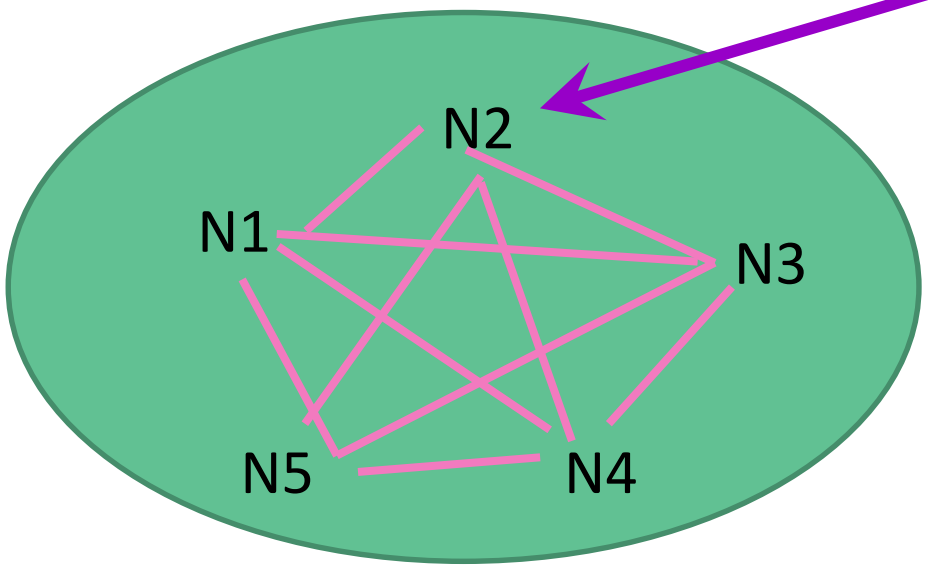
RSAConference2020

Private/Permissioned Blockchains

Group of approved ledger maintainers

- Still a chain of blocks, but no minimum hash value, so cheap to compute (or to forge)
  - Trust that most ledger maintainers are honest
- Maintainers have public keys, but usually only use it to create TLS sessions to each other
  - Because blocks are not signed, it's trivial to generate a false ledger
- Sometimes all the maintainers are run by the same organization
- Lots of strategies for writing a next block, e.g., choose a leader node who creates the next block
- Usually the ledger itself is not publicly readable -- Users just query for information
- People writing applications just think of "blockchain" as a black box with API for "store data" or "ask questions about the data"

Using a permissioned blockchain



Client asks a question

Believes what N2 says

What are the security properties of public blockchain?

- An attacker with more compute power can erase part of the chain
- Can’t record an invalid transaction (or your block will be ignored), but
  - Nothing prevents a node from discriminating against some transactions (e.g., those with smaller transaction fees)
- Subtle attacks possible by delaying messages
- If user queries one full node, that node can lie

What are the security properties of private blockchain?

- Permissioned, consortium of k independent organizations
  - Presumably some sort of quorum scheme among ledger maintainers, so need to subvert a quorum to overwrite history
  - If queries are to one node, then that node can lie. If write is to one node, it could ignore it.
- Private, or all run by one organization
  - Even with signatures, that one organization can completely change history
  - (positive security properties) ???
    - It's basically just an inconveniently formatted database
    - Allows you to say you're "using blockchain" and seem trendy

RSAConference2020

Centralized vs Distributed

Is centralized “bad”?

- Centralized: One organization in charge
- “Centralized” can, and usually does have
  - Lots of servers so application is always up
  - Store data in multiple places, geo-replicated (especially if using a public cloud)
- Centralized is the most efficient
- And it’s clear who to blame
- Most applications require “adult supervision” – someone to complain to if, for instance, merchant doesn’t ship the product
- So, most of the time, centralized is exactly what is needed
- For instance, when you withdraw $20 from an ATM, your own bank makes the decision, not a consortium of banks voting

What does “distributed” mean?

- Lots of meanings
- Store data in lots of places? (industry knew how to do that before “blockchain”)
- Have multiple instances of a server, to split load, and for resiliency? (industry knew how to do that before blockchain)
- Distributed trust…that’s interesting and subtle

Distributed Trust

- Any organization can become evil (evil employee, someone evil steals their private key and impersonates them)
- Byzantine failure: “Failing” by doing bad things (vs “halting”)
- Thousands of anonymous participants is always problematic
  - Reputation systems: Bad guy can create zillions of identities
  - A fascinating paper: “How a lone hacker shredded the myth of crowdsourcing”
- Does “blockchain” have “Byzantine robustness”?
  - Not really…51% attack, completely trust full node you query, no enforcement against a node refusing to record a transaction
- Are there any other ways of protecting against malicious participants? (next slides)

RSAConference2020

Distributed Trust without Blockchain

Triple Redundancy

- Simplest case of a voting system
- Three systems, answer is whatever two of them agree on
- Each operates independently…they don't speak to each other
- Data is input into all 3
- Answer is read from all 3

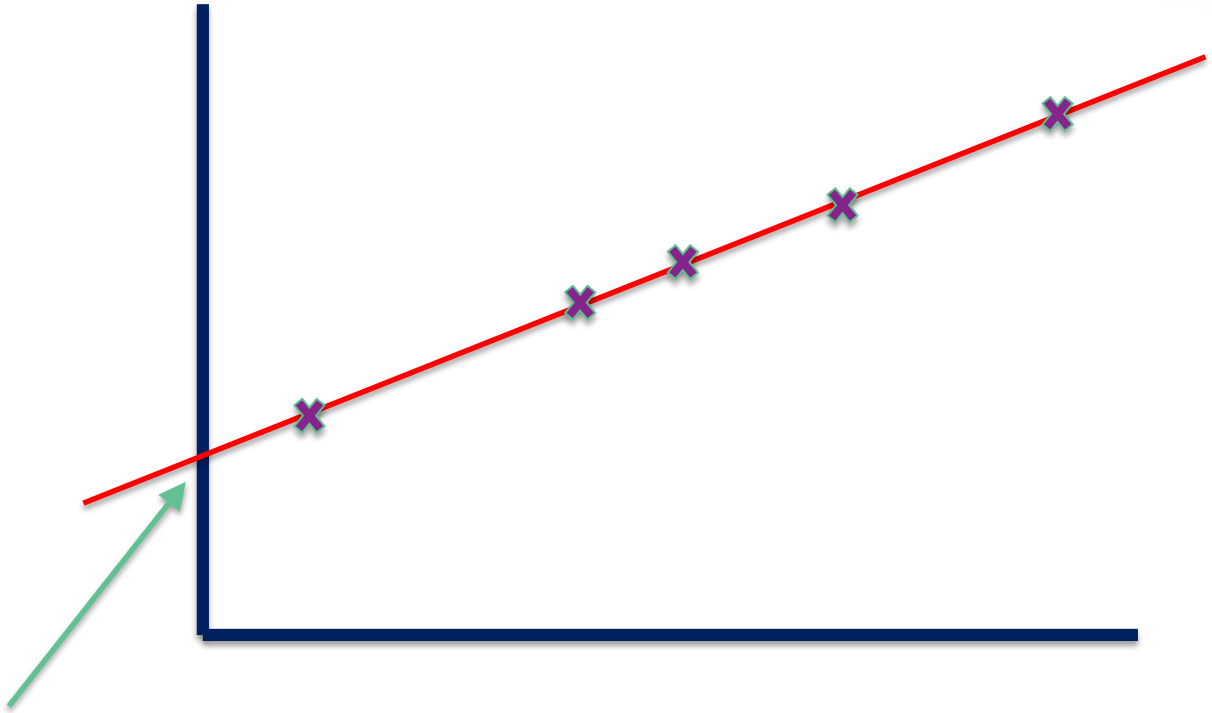
Example: Credit Rating Agencies

- Current method:
  - Anyone can be a credit rating agency
  - Perhaps with 3rd party accreditation agencies…get as many certifications as you want
  - Each operates completely independently, with their own sources of data, and algorithms for computing a score
  - Someone checking a credit rating can ask whichever they trust, or ask several
  - Plus laws that allow consumers to contest data they think is wrong
- With blockchain, presumably
  - Some consortium…who chooses the participants?
  - They'd have to argue about things, vote, etc., and it would be far less efficient
- Non-blockchain solution way more efficient, more democratic, more robust, more secure

Another example of distributed trust: Escrowing a secret

- If you're afraid you might lose a secret, you could give it to an escrow agent for safekeeping
- But suppose the escrow agent is not trustworthy, in one of two ways:
  - It might forget your secret
  - It might divulge your secret to others (or use it in an evil way)
- Solution: wonderful technology…Shamir's secret sharing
  - A secret is split into n pieces ("shares")
  - Each of n parties is given a share
  - Requires k shares to compute the secret…fewer than k gives no information
- This technology is simple and efficient, and forms the basis of many schemes that want to guard against some subset of participants being flaky

Shamir Secret Sharing, k=2



Secret S: where
line crosses y axis

- Solve for S in y=ax+S
- Given any two points, know the line
- One point gives no information
- For k>2, equation of degree k-1

Storing data using blockchain

- A blockchain does have lots of independent places that store the data, but…
  - How many places? Bitcoin's blockchain is estimated to have 30,000 full nodes. Imagine all the world's data stored in 30000 places
  - You can't delete data
  - And furthermore, a public cloud is contractually obligated to hold your data. Blockchain participants are voluntary and there have been many blockchains that ceased to exist
  - So, it's likely that the 6 or so locations that a public cloud stores your data in will be more robust (and way way way cheaper) than having a blockchain store the data in tens of thousands of places

Storing data

- Even if a cloud stores your data in lots of places, a malicious cloud administrator can delete all your data
- So, store your data in multiple independent clouds
- Detecting modification
  - Sign your data to detect modification of your data while in the cloud
  - Possibly get independent parties to also sign
    - If your private key were stolen, the bad guy can write whatever he wants and sign it with your key
- Integrity checks (signed hashes of the data) just detect modification…they don't help you recover corrupted data, so again…need to store your data with multiple independent organizations
- I miss mag tapes…offline…as long as system was uncompromised when data was stored, the data on the tape will remain valid

Notaries/Timestamping

- For certain transactions, it might be necessary to prove (beyond “trust us”) what occurred
- Also, be able to prove something happened before some time
- So, you can use a trusted (hopefully independent) 3rd party, a “notary” that signs (timestamp, hash) of the data
- What if the notary is bribed?
- Then get the transaction signed by several independent notaries
- These notaries can be very lightweight (just timestamp and sign the hash, and send it to the entity to store its own certified transactions)

RSAConference2020

Frequently Heard

Possible questions

- Typical questions people ask
  - What kinds of applications can I build on blockchain?
  - How can I apply blockchain to this problem?
- No!! Instead…
  - What problem am I solving?
  - What are various ways of solving it
  - Compare approaches
  - If ‘blockchain’ turns out to be the best solution, OK…

Misleading statements

- Look how many applications I can build on blockchain (with API of store data, read data)!!!
- Societal problem … E.coli … solution … record supply chain on blockchain!!!
- They could also, way more efficiently, use any storage system (disk, cloud, …)
- Doesn't answer any of the hard problems … credentials for farmers, RFID tags on lettuce leaves … and, a database would be more efficient

Misleading statements

- Look how much faster my application is now that I'm using blockchain!

- Governments can't regulate or outlaw cryptocurrency

- Sometimes delays are intentionally built in for legal reasons or safety…some human needing to OK a transaction bigger than some amount.

  Or perhaps, you are comparing some ancient process…all you're saying is that if you use computers and networks, it will be faster

- Yes they can.

  Maybe if you're careful, you won't get caught

  …just like with murder

Other Proposed Applications

Recording property deeds

- Who is trusted to do that? Building might burn down, or someone can be bribed to change the records
- What if they disagree?
- Why not just use a database?
- Use multiple independent registries
- Who gets to be THE blockchain? What if there are multiple blockchains?
- And what credentials do you use for recording information? Can I add a record that I own your house now?
- And what credentials do you need to prove you are the owner? A private key? What if you forget it?

Direct quote from blockchain article

- "Could blockchain be the answer to healthcare?"
  - "Imagine this: Your entire medical record is on the blockchain. Monitoring systems and IoT devices automatically update your data, so when you go for diagnostic tests, the results are recorded without a third party"

I can't imagine an application *less* suitable for blockchain

- World readable database?
- World writeable database? (who assigns credentials so you know data being written is traceable to the sensor that wrote it?)
- No organization to the data other than append-only log, mixing every human's records and sensor readings?
- No way to correct errors
- Data kept forever, in its entirety

I can't imagine an application *less* suitable for blockchain

- And yeah, I'm sure someone will think of using encryption to get around the world-readable thing
  - But with what keys?
  - Do you use trusted third parties?
    - Then why not have them manage the database?

Summary

- Start with “what problem am I solving”
- Then consider several types of solutions, and compare
- Don’t say “can we use this technology for this application”?
- What I say to engineers who are being pressured to use “blockchain”
  - Do the right technical solution
  - And then call it “blockchain”

RSAConference2020

Thank you!