Research Session 1: The Economic Limits of Bitcoin and the Blockchain

What impediments do cryptocurrencies face in becoming a trusted and widespread medium of exchange? And given rapidly increasing computing power, do they possess inherent advantages over traditional money? Some experts discuss and debate the future possibilities of bitcoin's role in the financial system.

Conference information

- Summary
- Papers, Presentations, and Audio and Video Recordings
- Speaker biographies

00:00 59:24

Transcript

Larry Wall: *I'm Larry Wall from the Atlanta Fed. As David [Yermack] explained in the last session, there's an important difference between the public, permissionless blockchains and the private, permissioned blockchains that were largely the focus of the last session. And I just want to emphasize that, really, blockchain is a family of technologies. And so they discussed—mostly, but not entirely—the ones where permission is needed to write to the blockchain. As they observed, this could lead to a closed club where the members who can write to the blockchain know each other and have some, but not necessarily complete, trust in each other.*

But most of the mind share, and a fair number of the questions, were devoted to blockchains that are used by privately created cryptocurrencies. Most of the implementations of these cryptocurrency blockchains strive to avoid reliance on what they call a "trusted third party." As a result, they theoretically permit anyone to write to a blockchain—but the flipside of allowing anyone to write on the blockchain is that the bad actors looking for cheap profits are among those likely to try to be writing to it.

The genius of bitcoin is that it specified a set of rules designed to prevent bad actors from being able to steal. In part—as was referenced earlier—these rules rely on cryptography to prevent one person from spending another's money. But another critical problem is preventing people from spending the same money twice. Bitcoin does this through what is called "proof of work," and that's what this session's going to be about. Proof of work has some limitations, one of the biggest being the low throughput for transactions, so some cryptocurrencies rely on alternatives.

But bitcoin is by far the most well-known and the most valuable of cryptocurrencies. Moreover, many people still regard the original bitcoin blockchain as the most secure—as one of the most, if not the most secure, blockchains in existence. Thus an important question is: Just how secure is bitcoin's proof of work? To answer this question, we're delighted that Eric Budish has agreed to present an important paper addressing the security issue. Eric is a professor of economics at the University of Chicago's Booth School of Business. We're also fortunate to have as a discussant one of the thought leaders in the Federal Reserve on blockchains and cryptocurrencies, David Andolfatto, who is a senior vice president at the Federal Reserve Bank of St. Louis. And with that, Eric.

Eric Budish: Thank you, Larry, and thank you to the Atlanta Fed—and to all of you for being here. It's a great honor to be able to present this research. This is a paper that makes a pretty simple economics argument about a pretty complicated computer science innovation, and you've had some preview to this computer science innovation in the last panel—which should be helpful—but what I'm going to do, the way I'm going to structure this talk is I'll spend the first slide giving you an overview of my argument. And then I'll spend the next several slides going over the computer science innovation at the level necessary to then get into the heart of my economics argument—which, as you'll see, is a pretty skeptical take on the Nakamoto blockchain.

So Satoshi Nakamoto, what he invented, or what she or they invented, from a computer science perspective was a new method of trust—specifically a way to generate trust in a dataset, often called a ledger, that's anonymous and decentralized—that is, it doesn't rely on a trusted party, doesn't rely on rule of law. And the method he invented is now what's known as the proof of work blockchains, proof of computational work, and it's this elaborate way of giving computers around the world an incentivized way to pay attention to the same dataset.

And what I'll show is that the amount of computational work generating this new kind of trust—this anonymous, decentralized trust—has to simultaneously satisfy two economic conditions. First is a zero-profits condition, or free entry condition, that the maintainers of the trust, the blockchain miners, earn an economic return on the computational work that they're putting in. And the second is an incentive compatibility condition, that the amount of mining activity is sufficient to deter what's known as a "majority attack"—that is, that the costs of an attack have to be larger than the benefit, otherwise the thing's going to get attacked.

And these two conditions together imply a third—which is pretty damning—which is that the recurring flow cost to miners for maintaining the bitcoin dataset have to be large relative to the one-off stock, if you will, benefits of attacking this dataset. And this is just very expensive as a form of trust. It's like a large, implicit tax. So the last part of my argument considers what I'll call the "sabotage scenario," which shows that if two things are true—so number one, the technology used to mine the bitcoin blockchain is specialized, which in practice it is, and two, a majority attack causes the value of bitcoin to totally collapse, and in practice it might because an attack might undermine trust in

this pretty novel and confusing thing. Then the cost of an attack is much larger. Instead of being the flow cost referred to above, the cost of the attack would be the stock cost of the specialized capital used to maintain the blockchain.

But this is a pretty damning concession in its own right, because it concedes vulnerability to sabotage and total collapse, and in fact, my analysis is going to point to some specific collapse scenarios that I'll go through in the end. So the overall take that emerges from my paper is that bitcoin—I think it's an ingenious computer science innovation, but I think it's unlikely to become an important part of the global financial system. It's pick your poison: either an extremely high usage cost or vulnerability to sabotage and collapse.

Okay, so as promised, I want to spend a few slides going over the basics of the Nakamoto blockchain innovation at enough detail of the level that you can understand the economic concerns about it that I'll raise in the rest of the paper. So a transaction you can think of—and I'll simplify some details, just to convey the main points—but a transaction you can think of as consisting of a sender, a receiver, an amount, and a signature. So Larry sends David 10 bitcoins signed by Larry. The signature uses traditional cryptography to prove the sender's identity and encode the transaction details.

So let's imagine that we accumulate transactions on a Google document, on a Google spreadsheet. So the signature already provides a certain level of trust, so only Larry can initiate a transaction in which Larry sends his money. I can't initiate such a transaction, because I don't know his cryptographic password, if you will.

But there are some obvious issues. He could send the same money twice. He could send money that he doesn't have. He could send money to David and then subsequently remove that transaction from the Google doc. So there's some pretty obvious stumbling blocks. Now let's imagine that transactions accumulate through a trusted party that keeps track of balances—this could be the Fed or a bank—and that actually works just fine regarding all of the security issues listed above. The trusted party could ensure that all transactions are funded, that the same money doesn't get sent to two people at once, and so forth, but it requires a trusted party, and the whole point of bitcoin and Nakamoto was to get around the need for a trusted party and the rule of law.

So what did Nakamoto come up with? So users submit transactions—again, sender, receiver, amount, signature—to what you can think of as a holding tank, a pending transactions list. And then this large mass of computer power that's anonymous and decentralized competes for the right to validate transactions one block of transactions at a time. So each new block of transactions chains to previous blocks—that's where the phrase "blockchain" came from—and for a block to be valid, several things must be true. First of all, each of the transactions in the block has to be individually valid, so it's got to be signed correctly and it's got to be funded—that transaction has to be funded, given history. And second, the set of transactions collectively can't contradict itself—so I can't send the same money to Larry and David in the same block.

This computational tournament is complicated, but you can think of it as like a search for—a quite elaborate search for—a lucky random number...really, a string of letters and numbers. And this lucky random number search is currently running at the rate of 50 million trillion lucky numbers checked per second—and even at that rate—again, 50 million trillion lucky numbers per second—it takes about 10 minutes on average, and in fact is calibrated to take about 10 minutes on average, to find such a lucky random number. This lucky random number is a function of the data in this block and the data in the block that it's chaining to, and the reason why it's called "proof of work" is it's incredibly hard to find, again, 50 million trillion numbers per second. But once I find one, it's trivial, computationally, for you to check that I have indeed found it.

A miner who finds such a lucky random number reports their new block. They report all the transactions in the new block, they report the lucky random number, if you will, that they found. Other miners quickly check that this block is valid—they check that all the individual transactions are valid, and the lucky number satisfies the hash puzzle. And then they quickly move on to working on the next block, and the successful miner earns a reward. They earn an economic reward, which you can think of as on the order of $100,000 dollars per block, per 10 minutes. Here's a visual of that—I'll skip this just because we've now talked about blockchain two sessions in a row, so hopefully by now the basics are conveyed.

So I want to describe, what does this elaborate construction accomplish? And I want to read from the abstract of the Nakamoto paper to describe what he's accomplished: the network timestamps transactions by hashing them into an ongoing chain of hash-based—"hash" is just the lucky number thing—proof of work, forming a record that cannot be changed without redoing all of the computational work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of computational power—as long as a majority of computational power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers.

So he's created anonymous, decentralized trust, so you can trust this dataset without any particular bank or legal entity behind it, because of this elaborate computational tournament. But it's vulnerable to a majority attack—it's in the abstract of the Nakamoto paper—it's not something I made up.

Okay, so now let me make one slide of clarification and then I'll go into the critique, which is: Larry said, "I'm talking about blockchain and the permissionless, anonymous, decentralized sense of Nakamoto, not what's sometimes known as a 'permissioned blockchain.' " A permissioned blockchain involves known, trusted parties. They don't involve the central intellectual innovation of Nakamoto. So here's a wonderful Matt Levine quote: "If you announce you're updating a database, no one cares about it—if you say that you're blockchaining the blockchain software used by a blockchain of blockchains to blockchain blockchain blockchains, the *New York Times* will blockchain a blockchain about it." *[laughter]* So my critique is not of distributed ledgers. My critique is of blockchain in the sense of Nakamoto.

Okay, so let me get into the economic critique, and it's really three equations—it's something you can teach undergraduates. So the first equation is a free entry condition for bitcoin miners. So let's let capital "P" be the reward for solving one of these computational puzzles—so think of that as $100,000. Let's let little "c" be the per block cost of one unit of computational power, so the cost of one lottery ticket in the elaborate computational lottery, if you will. Let's let "N" be the amount of computational power devoted to the lottery—so each lottery ticket has a "$1/N$" chance.

So economic free-entry equilibrium says that the amount of entries should be such where N times c equals P. So what does that mean? If N is much higher than that, then each lottery ticket isn't earning an economic return. The cost of the lottery ticket is larger than the one out of N chance that you're going to get this prize P. And if N is much smaller than that, then there's an entry opportunity—you should enter and add more computational power to the system. So that's the economic equilibrium.

The second condition is the incentive compatibility condition, that the system does not induce a majority attack. So what's the cost of a majority? Well, it's the same N^*c thing from the previous equation, on a per-block basis—so if there's N^* units of honest computational power, and I come at the system as an attacker with N^*+1 units of computational power, I now have a majority—and in my paper I'll simulate different scales of majority to try to get a feel for how long attacks would take in practice. Let's let alpha be the duration of an attack—so how long does the attack take?—and economically I want to net out the subsidy that an attacker gets, which is that they get block rewards over the course of their attack. And let big "V" be the value of a successful attack—and I'll go through some attack scenarios soon.

So the incentive constraint that the cost of an attack has to be larger than the benefits of the attack is just this equation two, where alpha times N^*c has got to be bigger than the value of an attack. So what the issue is if you put these two equations together—and you notice N^*c, the per block cost of computational power, showed up in both of them—they imply a third equation which says that capital P, the prize per block, has got to be large relative to the value of a successful attack on the bitcoin blockchain. So in words, the equilibrium per-block payment to miners for maintaining this dataset has to be large relative to the value of attacking this dataset, or the flow payment to miners has got to be larger than the stock value of an attack.

And this is just a very expensive form of trust. The reason why it's so expensive is it's memoryless—the system is only as secure as the amount of computational power devoted to it right now, in these ten minutes or this hour. Most other forms of economic trust are a lot less memoryless: relationships, or brands, or trust that emerges from laws and force. Imagine if to attack the Fed, you just needed to have more computational power than the Fed for 10 minutes—it's just a very weird trust model.

Then from a security perspective, the critique is that the security model is extremely linear in the scale of attack. A $1 billion attack is a thousand times more expensive to secure against than a $1 million attack. A $1 trillion attack is a thousand times more expensive to secure against than a $1 billion attack, and so forth. Most other forms of IT security have much better economies of scale, because they involve traditional cryptography, not 51 percent majority. They involve force, rule of law. Again, imagine if you were attacking the Fed and you needed to just have more computers than the Fed for a couple hours.

Okay, so I now want to talk about the possibilities for attack, to try to flesh out the economics. So let me spend a slide clarifying what an attacker can and can't do: so an attacker can solve these elaborate computational riddles faster than the honest minority. This allows the attacker to create an alternative longest chain, replacing the honest chain at a strategically opportune moment, which allows the attacker to control what transactions get added to the blockchain and remove recent transactions from the blockchain. The attacker also is earning rewards during their attacks, so they're getting subsidized.

What a majority attacker can't do is "steal all the bitcoins." For that I would need to not just have 51 percent of the computational power, but I would need to break modern cryptography. So the canonical attack to worry about is called "double spending," so in a double spending attack, first the attacker spends bitcoins—so I send bitcoins to, say, a financial institution in exchange for some financial assets. Two, I allow that transaction to be added to the blockchain, and then three, I subsequently remove that transaction from the blockchain—maybe after an escrow period, after I've gotten whatever assets I've spent the bitcoins for. So now I've spent the bitcoins, but I've recovered them, or I've taken them back, and I have whatever assets I've spent on, so I can "doubly spend" those bitcoins.

Under some assumptions—which I detailed in the paper in a lot of detail—you can translate that equation from before, to this very similar looking equation that's on a per-transaction basis, where the left hand side is the price per transaction, the fee per transaction, paid to bitcoin miners, and on the right hand side is the value of the transactions that can be supported by the bitcoin system without inducing an attack—so V transaction is a statistic on the largest kinds of transactions feasible using the bitcoin blockchain.

I run some computational simulations in the paper—and I should add, by the way, that under slightly different assumptions this equation can be a bit too aggressive or a bit too conservative. I run computational simulations in the paper with an escrow period of six blocks, which is very common in practice in bitcoin these days. That's about one hour. The level of alpha you would need to not induce an attack is a little over three, which translates to about a 30 percent implicit tax rate to not induce attacks. Even with an escrow period of 1,000 blocks, which is about a week, the implicit tax rate would be a few percent—and this is a percentage of the largest feasible transactions, because remember: if you're going to attack bitcoin, you're not going to attack it $100 worth of opiates at a time. You're going to go after $100 million at a time from financial institutions.

So just to illustrate double spending, in case that's not all clear—so let's say the last block before the attack is on the upper left. Attacker sends bitcoin to banks, exchanges, and so forth. There's an escrow period. After the escrow period banks, exchanges, and so forth release their assets to the attacker. In parallel, the attacker is mining his own private chain. In that private chain, he sends those same bitcoins elsewhere, so they can never be recovered by the banks or exchanges. With certainty, if he has the majority his chain will eventually be longer than the other chain because he's got more computational power. He can solve lotteries faster than the other guys. At a strategically opportune moment, after the assets have been released and when his chain is longer, he releases his chain and that then becomes the official record.

Okay, so some takeaways from the double-spending simulations: it's consistent with early use cases of bitcoin. It's more skeptical of financialized uses of bitcoin, like to store a value story or a global payment system in size. For the system to be secure for large transactions requires fees that are likely to render it prohibitive for smaller ones, and I guess the surprise to the computer science community is that the escrow period isn't more protective. And economically, the reason for that is that the attacker is getting subsidized by block rewards—so even a 1,000 block escrow period, the attacker is going to get back a 1,000 block reward that subsidizes the cost of attack.

Okay, so now I want to talk about the possibility of a sabotage attack. So if two things are true, then the cost of an attack is larger than this flow cost. So number one, the mining technology is specific to bitcoin—and in practice, it is. These bitcoin-specific ASIC [application-specific integrated circuit chips] chips are extremely fast, such that repurposable technology is not competitive. And number two, the attack's a sabotage that causes a dramatic decline in the value of bitcoin. Under these two circumstances, the cost of an attack is significantly larger, so just to give you intuition, consider a 100 percent collapse—an extreme case: double spending is now pointless. I send you a $1 billion of bitcoin, you send me a $1 billion of financial assets—I double spend, I now have both the $1 billion of bitcoin and the $1 billion of financial wealth, but the $1 billion of bitcoin goes to zero so I haven't really netted anything.

So you see, now the cost of the attack becomes the stock cost of the capital, right?—because now I'm left with worthless bitcoin, and I'm left with worthless bitcoin-specific mining technology, these mining rigs. So that yields a new economic constraint, which is that the stock cost of capital has to be larger than what I'll call "V sabotage," the value of sabotaging and bringing the whole system down. And that's now a few billion dollars, not a few hundred thousand or a few million dollars, so that's much more significant.

But it's a bit of pick your poison, right? To get this constraint you have to concede the possibility that an attack is going to bring the whole system down, then you have to worry about an attacker motivated by bringing the whole system down. So you either have to concede high implicit tax rates, or risk of collapse. Neither are that appealing, in my view.

So suppose I'm right that the reason bitcoin hasn't been attacked to date is this flow versus stock argument—that is, if you could attack it using a flow cost of capital attack model, someone would have attacked it by now. But because of this combination of the specific capital, and an attack would bring the thing to zero, it's too expensive. That then implies some scenarios for when the calculus on an attack might shift, and an attack might become in an attacker's interest.

So scenario one is just conditions change in the chip market, in the specialized technology market, and this could happen in two ways. One is the technology matures, and then the chips get cheap enough that electricity becomes the predominant component of cost. Or second is if the bitcoin value falls for other reasons, and then there's a glut

of chips—that could also lead the predominant component of cost to be electricity as opposed to the capital. So under either of these conditions, the cost of attack's much lower.

Second, if repurposable chips catch up to specialized chips—a similar argument. And then third, if economic sabotage becomes sufficiently tempting to justify the few billion of expense rather than a few million.

So 51 percent attacks have started to materialize in the last year or so—mostly since my paper first circulated. Not causally—I'm not a 51 percent attacker, don't worry. The one I'll call—that joke may not have landed *[laughter]*—the one I'll call attention to is bitcoin gold, which is...it's both the largest and maybe best-known, but I think the economic point this case study illustrates is that the financialization induces attack. So bitcoin gold got attacked through exchanges, so the attacker was able to extract $18 million, at least, worth of other cryptocurrencies from exchanges and then undo the bitcoin gold transactions. Attacking to extract wealth out of financial institutions scales a lot better than attacking $100 or $1,000 worth of goods off the dark web at a time.

So just to conclude: Nakamoto is a genuine intellectual whose discovery created something that didn't exist before—anonymous, decentralized trust in a payments ledger. It's ingenious, I think, but economically quite expensive. Equation three says that for this trust to be meaningful, the flow cost of running the blockchain has to be large relative to the one-shot value of attacking the thing. So it's as if there are payments to Visa every 10 minutes, or a large—relative to the value—of a terrorist attack on the Visa network. So this is like a large implicit tax on the use of bitcoin as a payment system.

The argument that an attack costs more than this flow cost requires one to concede two things: one, that the security rests on very specific capital—these bitcoin ASIC chips. And two, that if someone attacked, bitcoin would collapse. It would be interpreted as a sabotage attack. So it's pick your poison: either vulnerability to outright collapse, or high tax rates—and the analysis points to specific collapse scenarios, especially as conditions change in the chip market or if bitcoin's value falls enough. The overall message, and takeaway of the paper, is there are just intrinsic economic limits to how economically important bitcoin can become. If it gets important enough, it will get attacked, and that suggests to me that it won't get that important.

My last remark is: I want to emphasize that the models consistent with the early uses of bitcoin and blockchain, I'm skeptical—as I hope is clear—about its use in more financialized contexts, like store value or a major component of the global financial system, but I'm not skeptical of the use of distributed databases more broadly. We can get into a Talmudic dispute over whether a distributed database, under what circumstances it should be called a blockchain or a permissioned blockchain or whatever. But I'm not skeptical of the use of distributed databases or ledgers. And indeed, what my paper highlights is it's exactly what's so exciting, from an intellectual perspective, and so innovative about Nakamoto: this anonymous, decentralized trust that emerges from proof of work—that's ultimately, it's economics' Achilles' heel that will make it, I think, limited in power. And let me stop there. Thank you very much. *[applause]*

David Andolfatto: Good morning, everybody. First I'd like to thank the Atlanta Fed for hosting this conference and inviting me to participate. I've been really impressed, I have to tell you—I've attended a lot of conferences on bitcoin and blockchain over the years—since 2014, at least—and I was studying it even before that. And I think I've seen some of the most sober analysis I've seen to date. So what you're getting is really, really good stuff—the previous session, and Eric's contribution as well.

So my history is as an academic—a monetary theorist, in fact—and so I became interested in this project, bitcoin and blockchain, very early on, and like many people, I was very skeptical. But once I looked under the hood, I became intrigued. I saw something really interesting there. And over the years my thoughts have evolved, but they've largely evolved, I have to confess, basically, to Martin and Keith's views of the previous session—kind of a healthy dose of skepticism, I think, although I admit there might be some niche uses for the innovation. So what I thought I'd do for this talk is just to give you an overview, I guess, of an academic's perspective of what the endeavor is about, and then I'll get specifically into Eric's talk, and give my brief assessment of it.

At the end of the day, what we're talking about here is database management. It's really basically accounting—and if you put it that way it doesn't sound very sexy. It doesn't sound very exciting, but in fact accounting is tremendously important in societies. And the type of information that we're accounting for, or keeping track of, is kind of specific. It's information, I would say, that relates to individual action histories—things like credit histories, where did you go to school, what degrees did you attain, if you're a company what is your customer satisfaction record, are you a supplier that delivers supplies on time, what is your history?

This type of information economists label as "intrinsically useless information." It's intrinsically useless in the sense that if you were stranded on a deserted island, it would be very difficult to subsist off of your credit history. Nevertheless, this credit history—this information—takes on value in a societal context: firms, businesses will devote a considerable amount of resources to acquire your credit history. But because this is just basically information that can be fabricated out of thin air, and because this information is valued, it basically can be...it takes the form of a currency: your reputation is something that you can use to get stuff, to get favors from people. People can use their reputations to take command over resources and so reputation. What is reputation? It's just information about your individual past contributions to society, more or less.

And so because this type of information takes on this property of currency, it's going to elicit all the familiar incentives that are associated with currency. They're going to be great incentives to counterfeit this information, to fabricate it, or to steal it. And so the key question that any community has to face is how a community wanting to share and manage this type of information relating to the various contributions that its members make to the community, is how to manage such information when trust is lacking, when there's evil in the world, when people are willing and able to fabricate this information to claim that you did something bad when you didn't do something bad—or to claim that they've done something good when they didn't do something good, to boost up their reputation—and perhaps destroy a competitor's reputation.

Historically, small societies have relied on communal models and larger societies on delegated models. I have written a blog post on this, about this spirit of blockchain. A lot of people find blockchain very mysterious, and I think justifiably so, given the way it's described. But what I recognized in the spirit of blockchain is something very ancient, something I've called the "primitive blockchain," and that has to do with the way information in small communities are managed and shared and updated. And so, famously, anthropologists have gone and studied primitive societies and noticed, for example, the absence of monetary exchange, and the question is, "How did they manage the bookkeeping? How did they keep account of who has done what?"

Well, the answer is in small communities—and you might be familiar with how these small communities work, because we all belong to them: families, friends, people at work—we keep track of each other's actions in our head, the computer servers that live in our head, and we communicate. We communicate with each other, and come to some sort of shared consensus on what our shared history is in this community. And this is the way small communities function, and also to this present day. It's something that [former Minneapolis Fed president] Narayana Kocherlakota labeled "societal memory." So Narayana Kocherlakota is a former president of the Minneapolis Fed, but also a very well renowned monetary theorist.

So what are the problems associated with this sort of communal model? Well, there's obviously the incentive to fabricate information to spread rumors. How do you guard against that? Having an immutable ledger, when somebody spreads a rumor about yourself and it spreads in the community of your friends and is accepted communally and

becomes part of the shared history of the community you belong to, but it's wrong, and immutable—this is not necessarily a desirable property. I think that Martin was pointing this out. Nevertheless, these types of communal models kind of work pretty well. They work pretty well in small groups. The problem with them is that they don't scale, and so this is why we have delegated models that permit record keeping to scale.

And the question here is whether or not recent innovations in computer storage, electronic communications, cryptography, game theory, can now permit this communal method or this communal protocol to kind of scale in some manner, and I think that's what this blockchain spirit is all about, and that's kind of an exciting idea: the idea that you can dispense with these trusted third parties that are very powerful—which is fine if they can be trusted to use their power in the interest of society, but that's not often the case.

But what is a blockchain, exactly? I think the way I define it is it's just a database management system, with the following properties: it has hash-linked data structure with an open-read privilege. I think the open-read privilege is not a big deal. If I go to any public library, there's a database of books that has an open-read privilege—I just go and there's the gatekeepers there—the librarians, of course—but open-read is nothing defining, I think, about blockchain.

I think it's the second property that is the more defining characteristic, the write privilege—who gets to write history? I mean, who gets to write the history books in the library? Do you trust an accredited historian and the librarians to manage the database? Or is the outcome more open, kind of more consensual? And in the case of blockchain—or bitcoin, I should say, proof of work in particular—the outcome is the consequence of an open, noncooperative game with no legal recourse. This is in stark contrast to conventional database management systems—right?—where data structures are more general, but with restricted read privileges and permissioned access. And the write privilege is restricted to, and delegated to, a legally liable third party.

So then, having described what a blockchain is, then the natural question is: Why a blockchain? Conventional database management systems, especially large scale ones, are inherently more efficient, I would say. A cheating consensus has got to be more difficult than delegating it to a single decision-making body. And so I like to compare FedCoin to bitcoin, for example—or Fedwire...did I call it FedCoin? *[laughter]* I did have a blog post promoting the idea of FedCoin as well, but Fedwire is the interbank, real-time gross settlement system that clears payments across banks in the United States. It processes about $3 trillion a day, and it costs basically nothing to run—especially in relation to the volume that's transacted. It's very cheap.

And this shouldn't be a surprise. We were just talking about accounting, it's just debiting and crediting accounts—and that's not conceptually a very difficult thing to do. You have to worry about security, and all that sort of stuff. But compare that to bitcoin, which is intrinsically much more expensive, and I would say any consensus-based mechanism has to be inherently more expensive—it just has to be. I think that that should be a theorem. I'll write it down and prove it for you.

But then nevertheless having admitted that added expense, why might a blockchain be preferred? It might be preferred if, for example, a delegated record keeper is either not trusted—we're all familiar with the high-profile data breaches of Yahoo and Equifax, for example. We all are aware, I know the central bankers in here, of the distrust of the Fed and of the banking system. Perhaps you just don't trust these institutions—okay, fair enough. Then go blockchain.

Or perhaps it's not an issue of trust. Perhaps it's just an issue of an agency exploiting its monopoly power, like Western Union in the case of international remittances, and you want to bypass that monopoly power—okay. Or it might be, for example, you belong to a community of firms in a supply chain that would like to share and manage information among different firms in a supply chain. You'd like to delegate this responsibility to a third party, but it's just not available—again, that might be a case where this kind of communal structure might be an alternative way to manage and share information.

But the question is, can this proof-of-work-based blockchain scale? This is, I think, the question. And the hope for a very long time has been "yes," but Eric's paper—this is something...I've been studying this for a long time, and I have to say I've never thought about the question that Eric raised here, so I thought it was a very interesting contribution. And Eric says that the answer may very well be "no, fundamentally no"—at least, theoretically. So we sometimes joke that economists are people who see something working in practice but argue that it can't work in theory...so theoretically, bitcoin can't work.

The argument—let me just go over it very quickly: if you want to think of P as the prize of a lottery, suppose you want to buy a lottery ticket and the prize is P. This is the block reward for the bitcoin miners. Think of the miners—the "miners," that's a misnomer. The miners are accountants, they're managing the books. They're part of the process of managing the books, and they have to be paid. They have to be paid wages, just like any accountant at any bank. They're paid in a particular way, they're paid kind of randomly, they have to win a...you show up to work, you do some work, and you may or may not get a paycheck—but when you do get the paycheck, it's a whopper. That's P.

Well, what determines the probability that you get paid? Well, you can buy lottery tickets. The more lottery tickets you buy, the more likely you're going to win the lottery—and each lottery ticket costs "c." And so in economic theory, then, if we fix the prize and we fix the cost, and we ask, "How many lottery tickets can we expect to be sold?" The answer is: "Well, we'd expect the number of lottery tickets to be sold—N^*—to be such that the marginal benefit and cost of buying a new lottery ticket is just equated," so you're just indifferent between buying that last lottery ticket or not. So that N^* is representative if you want to think about how many miners are in the system willing to bear the expense of keeping the books.

Now one of the striking things about this property is that if you increase the wage rate, the block reward in blockchain, all you do is you induce more entry: people just want to buy more lottery tickets. And that's costly. It's computationally costly—there's the hardware you have to buy, the electricity—and so increasing the block reward increases the expense of maintaining the ledger. And remember: the ledger is fundamentally...to maintain the ledger, the cost of that is basically zero—not quite. This is debiting and crediting accounts. I can do all of the bitcoin transactions on my laptop. It's just debiting and crediting an account. So this expenditure is in some sense kind of a waste in some way. It's the cost that you have to be willing to bear to have this consensual, decentralized recordkeeping system.

So for proof of work, the majority attack is going to be linear in the number of lottery tickets that have been sold. Just think about it very intuitively. Suppose you want to have a better than 50 percent chance of winning the next lottery. You're going to have to buy a lot of lottery tickets if it's a very popular lottery, and you won't have to buy that many if it's not a popular lottery. So the "no attack" condition requires the condition that Eric stipulated, that the expected cost of a majority attack has to exceed the value of attack V. Fine, so then what determines V? This is kind of an open question. What determines V in this scenario could be the largest value transaction that's taking place on the blockchain in bitcoin. So you can imagine transactions of $5, $10, $100, $1 million, $10 million.

The attackers are going to attack the largest value transaction, as Eric explained. Why go for the small potatoes? Go for the big $100 million transaction, and take advantage of that. Basically, spend $100 million twice. Or the value to an attacker could be through just sabotaging the system—perhaps litecoin, for example, wants to send attacks to take down bitcoin to increase the value of their own network.

So the point that Eric raises that I never realized before is this number could be very large—very large. And if you look back at the condition, the no-attack condition, and you combine it with how the cost of computing is related to the wage rate that you're paying these miners, then the no-attack condition means that you have to be paying these miners a lot of money, every day. In fact, every ten minutes, you have to pay them a lot of money to keep the network secure at every moment—every ten minutes. And that's just to guard against, say, this one-time attack of $100 million, or one-time transaction of $100 million. You'd have to do this on a daily basis to prevent that attack, and all the individual smaller-value transactions have to pay this higher fee, because this is the fee that's required to prevent the attack from happening. This is not a problem that conventional database systems have if they're well designed, because they're based on identifiable, legally liable third parties—so they're less susceptible to this problem. So I thought it was a very interesting insight, one I hadn't thought about before.

A couple of the questions I have here is...bitcoin is not something that's static. Bitcoin is a code, an open-source software, and it is an evolving code. It's not just written in stone. The developers add patches to it, the code evolves over time—very much like the U.S. Constitution is a code that becomes amended over time to suit the needs of the community, as the needs of the community change. And so we have to take into consideration that the code might evolve over time to bypass some of these apparent deficiencies, these theoretical deficiencies. One possibility might be to make the prize, how much you're paying the accountants, contingent on the maximum transaction size. So normally you're paying the accountants $20 an hour for regular processing transactions. Every once in a while $10 billion wants to be transferred, and then you up the prize and you draw in a larger amount of computing power and make a majority attack more costly. Is that a possibility?

The other possibility is that if scaling can't occur within the protocol, is it possible. Eric, I'm wondering if the scaling might occur along the extensive margin, so that you have many different bitcoins, litecoin, bitcoin 2, bitcoin hash. We can perhaps see the scaling of this endeavor occur—not within a protocol, but along the extensive margin. So if I wanted to send $100 billion, the way I would do it is I'd send $10 billion on bitcoin, $10 billion on litecoin, etc., something like that.

The analysis seems targeted at proof of work consensus protocols—that's the feeling I got when I was reading the paper—and I'm wondering: is this a generic weakness, in your view, of decentralized consensus mechanisms? Or is this something more specific to proof of work? Because there are other consensus protocols out there, proof of state being the most famous one, the nearest competitor. So I'm wondering: if so, is decentralized recordkeeping kind of doomed to fail? Thank you very much. *[applause]*

Wall: *So with the slide that was just there—and there are a couple of related questions that get at the issue of alternative consensus mechanisms. Here's a statement about proof of work as it is now: "The space is constantly innovating the protocol. Will this limit always be true?" And "doesn't this paper highlight the big incentive to find an alternative consensus mechanism?" They're kind of all getting at the same issue, and so how would you react?*

Budish: I think these are all...so David, thank you for the great discussion and for posing those questions at the end. But I think it's a question raised by the paper, so the first thing to say is, if you're a graduate student—which, this isn't a room of graduate students, but it's a research topic I push graduate students to push on. Let me kind of say a few related things—I'm speaking to this question, and the ones from David at the end. So one on modifications to bitcoin to make it more secure: the bitcoin community is working on that. They're working on responses to my paper. I'm open-minded. We'll see if...one in particular that I think's kind of in the spirit of what David suggested on his last slide of scaling with P, is the idea of a second layer. So the second layer idea for bitcoin is just concede that transactions on bitcoin are intrinsically really expensive, but try to do a lot of transactions off of the blockchain, that very periodically net on the blockchain, as a way to conserve on costs. I think that seems intellectually interesting, and plausible.

With respect to other consensus mechanisms, I think that the basic logic of the paper to me seems fairly general, in that there's got to be some economic incentive to do the validation, to be the accountants, as David put it—whether you're in a proof or work system, where the cost is computational work, or a proof of stake system, where the cost is you've got to post some stake, which has economic opportunity cost, the amount of validation effort you're going to get depends on what you pay for it. So some version of equation one seems pretty general.

And then some version of equation two, that if it's an anonymous decentralized system, what's preventing it from being corrupted or attacked? Some version of that seems fairly general. So I think what's most interesting about alternative consensus mechanisms is not getting around the basic logic of the paper but making the value of an attack smaller, and in particular because stakes have memory. Work's very memoryless. My computational work's only what I've done in the last ten minutes, whereas stakes have memory, they could build reputations. I think it's plausible that there'll be a proof of stake model that's more resistant to attack than the proof of work model, but it's kind of wait and see. I think it's an intellectual open question. And maybe some graduate student will prove a theorem that there is no such possibility.

Wall: *So, yes—by "proof of stake" you mean that some people have put up some money, or put up some of the cryptocurrency, and said, "I want to be one of the ones validating transactions. If you give it to me, I get paid a little bit—not very much, because I'm not doing a lot of work—but if I cheat, all that's at risk.*

Budish: Right, so it's sort of like owning shares in a...it's not exactly an analogy to owning shares in a corporation, but it's got some of that feel, like your ownership stake, your ability to validate scales with your ownership stake. But it's vulnerable to the 51 percent attacks. It creates new intellectual problems that aren't present in proof of work, that the computer science literature has been spending a lot of effort on.

The CS folks that I trust say the jury's still out on whether there'll be an alternative consensus mechanism that responds to the critique in my paper. But I'm open-minded—we'll see. I don't think you should have a trillion dollars of market capitalization hinge on an intellectual open question, but I'm not a valuation expert.

Wall: *So let's deal with a question just about how bitcoin works. We talked about a high implicit tax, but transactions could be as cheap as 30 cents, compared to 3 percent for Visa or Mastercard. So are you really paying a lot on bitcoin?*

Budish: So the fee, in practice, is predominantly seigniorage. So the payment to miners is 12.5 bitcoins per block—which at today's prices is $95,000. And then in addition to that, small transaction fees. And those transaction fees—and that's like, I extracted from this in my talk, but it's a fee to jump the queue, if you will. It's an inducement to put my transaction on the block sooner out of that that kind of holding tank, and those usually are quite small, as this question points out, so the P in my paper includes the seigniorage cost, which is part of what's economically inducing miners to validate. If users of the bitcoin system think that that seigniorage cost doesn't cost them anything, then...I mean, it's not economically right, but that's not for me to say.

Andolfatto: That's the same as when we go and purchase a product and we're not charged the 3 percent. It's the merchant.

Budish: Right, and it's implicit—exactly.

Andolfatto: The cost of the good is implicit in the fee, so you are paying it in that manner.

Wall: *Yes. And it was noted earlier, that seigniorage is going away. It's being reduced through time, and it's going to go away. And I think there's an interesting paper from the Bank for International Settlements that analyzes, though, what will it mean when it goes away, and it raises similar sorts of concerns, too. How about this one: If central banks move to running distributed ledgers, aren't there increased risks to geopolitically motivated attacks, and what are the inherent costs?*

Andolfatto: Well, I guess I'm the central banker, so...It depends very much on what one means by "distributed ledger." I think it was described in the previous session, we have to make distinctions. As "distributed," does that just mean an open-read database, like a public library? So I actually don't see why central banks would want to move to that model in the first place.

But suppose they did. Would there be increased risk to geopolitically motivated attacks? Well, that depends on who's doing the bookkeeping, right? So again, distributed ledger by itself does not imply anything about what mechanism you have in place to actually do the accounting. So the security is going to be determined by what protocols you have in place—the normal sort of database management protocols in case. Distributed ledger to me just means you're opening the ledger to be visible—in this case by distributing it, but even if it was just living on a platform and you can read it like on the internet, to me that's a distributed ledger. I'm not sure I know what the rationale would be for going to that model, though.

Wall: *Yes, and it's also certainly the case that central banks are already being subject to cyberattacks, as the Fed's internal security people are constantly reminding us. We've got just a few seconds, and I think the really big open question that we've talked a little bit about is whether alternative models can work better. And so let me go to a question we just recently got: In proof of stake, doesn't it put the control back into the hands of those with capital to use as proof of stake, and this is a way back to trusting the big banks?*

Andolfatto: I can take that last question there. It says: "Isn't it getting back to trusting big banks to process transactions?" This issue about—I think it was Martin who brought it up in the last session. I think trust is not an issue here, I don't think at least on the wholesale. If you look at Fedwire, for example: Fedwire is run by the Fed, and it's very efficient. It transfers funds for banks—trillions of dollars a day, and it's like: why wouldn't you trust the Fed? The Fed's not going to steal your money. This is for the reporters in the room: the Fed's not going to steal your money. How do we know that? Because the Fed can print all the money it wants, it's not going to steal your money, so *[laughter]* why wouldn't you trust the Fed?

Budish: You always want to ask where trust is coming from, right? So, trust: my argument is about shedding some economic skepticism on anonymous, decentralized trust as envisioned by Nakamoto. Centralized forms of trust—I'm reasonably happy with living in a society with laws and force and entities like the Fed and banks. But a lot of people aren't, and that's a bigger intellectual question that the craze over bitcoin and cryptocurrencies has brought a lot of attention to. But what I hope comes out of my paper is that people ask: Where is the trust coming from? And if it's coming from anonymous, decentralized, Nakamoto-style consensus, that's not free—that's vulnerable to majority attack. If it's coming from more traditional sources, like the Fed or rule of law, that it brings with it its own issues—issues that I'm reasonably comfortable with, but I guess I'm privileged. And I'll leave it at that.

Wall: *Great. With that, I think our time is over with, so let's thank Eric and David. [applause]*

RELATED LINKS: mp3 • • Past Financial Markets Conferences