

High-Level Debate

**Bitcoin Blockchain for Distributed
Clearing: A Critical Assessment**

Robert Sams

Journal

THE CAPCO INSTITUTE JOURNAL OF FINANCIAL TRANSFORMATION

RECIPIENT OF THE APEX AWARD FOR PUBLICATION EXCELLENCE

INDUSTRIALIZATION AND TECHNOLOGICAL INNOVATION IN FINANCE

Download the full version of The Journal available at CAPCO.COM/INSIGHTS

FIS IS YOUR COMPETITIVE ADVANTAGE

for an industry that is in constant motion.

Your trusted strategic partner worldwide

FIS™ is proud to be named the No. 1 financial services technology provider for the fourth consecutive year. Our focus is solely on understanding and serving your unique needs and helping you thrive in times of unprecedented change. Putting our clients first is key to continued success.



Journal

The Capco Institute Journal of Financial Transformation

Recipient of the Apex Award for Publication Excellence

Editor

Prof. Damiano Brigo, Editor-in-Chief of the Capco Journal of Financial Transformation and Head of the Mathematical Finance Research Group, Imperial College London

Peter Springett, Managing Editor

Sam Price, Assistant Editor

Head of the Advisory Board

Dr. Peter Leukert, Head of the Capco Institute, Head of the Editorial Board of the Capco Journal of Financial Transformation, and Head of Strategy, FIS

Advisory Editor

Nick Jackson, Partner, Capco

Editorial Board

Franklin Allen, Nippon Life Professor of Finance, The Wharton School, University of Pennsylvania

Joe Anastasio, Partner, Capco

Philippe d'Arvisenet, Adviser and former Group Chief Economist, BNP Paribas

Rudi Bogni, former Chief Executive Officer, UBS Private Banking

Bruno Bonati, Bruno Bonati Consulting owner and Chairman of the non Executive Board at Zuger Kantonalbank

Géry Daeninck, former CEO, Robeco

Stephen C. Daffron, CEO, Interactive Data, former Global Head of Operations, Institutional Trading & Investment Banking, Morgan Stanley

Douglas W. Diamond, Merton H. Miller Distinguished Service Professor of Finance, Graduate School of Business, University of Chicago

Elroy Dimson, Emeritus Professor of Finance, London Business School

Nicholas Economides, Professor of Economics, Leonard N. Stern School of Business, New York University

Michael Enthoven, Board, NLFI, Former Chief Executive Officer, NIBC Bank N.V.

José Luis Escrivá, Director, Independent Revenue Authority, Spain

George Feiger, Former Executive Vice President and Head of Wealth Management Zions Bancorporation

Gregorio de Felice, Head of Research and Chief Economist, Intesa Sanpaolo

Hans Geiger, Professor Emeritus, Department of Banking and Finance, University of Zurich

Peter Gomber, Full Professor, Chair of e-Finance, Goethe University Frankfurt

Wilfried Hauck, Chief Financial Officer, Hanse Merkur International GmbH

Pierre Hillion, de Picciotto Chaired Professor of Alternative Investments and Shell Professor of Finance, INSEAD

Thomas Kloet, Former Chief Executive Officer, Retired at TMX Group Inc.

Mitchel Lenson, Non-executive Director, Nationwide Building Society

Donald A. Marchand, Professor of Strategy and Information Management, IMD and Chairman and President of enterpriseIQ®

Colin Mayer, Peter Moores Professor of Management Studies, Saïd Business School, Oxford University

Pierpaolo Montana, Chief Risk Officer, Mediobanca

Steve Perry, Chief Digital Officer, Visa Europe

Derek Sach, Head of Global Restructuring, The Royal Bank of Scotland

ManMohan S. Sodhi, Professor in Operations & Supply Chain Management, Cass Business School, City University London

John Taysom, Visiting Professor, Department of Computer Science, UCL Policy Fellow, University of Cambridge, CSaP Board, The Web Science Trust

Bitcoin Blockchain for Distributed Clearing: A Critical Assessment¹

Robert Sams — Founder and CEO, Clearmatics

Abstract

There has recently been a dramatic increase in interest in the idea of using distributed consensus technology to facilitate the settlement of financial transactions. One strategy that has been advocated attempts to use the Bitcoin blockchain, running meta-protocols on top of that network, so that off-chain assets such as securities and property titles can leverage the same transaction protocol used by the endogenous on-chain cryptocurrency asset. This article explains Bitcoin's unique flavor of distributed consensus algorithm (hash-based proof-of-work) and how it was motivated by a design

goal of censorship-resistant digital cash. It then shows that censorship-resistant consensus has no mechanism for enforcing the correspondence between blockchain reality and legal reality that off-chain assets require. It also suggests that the security of the Bitcoin network itself would be compromised by such an attempt.

¹ This is an edited version of a blog that originally was published at <http://www.clearmatics.com/2015/05/no-bitcoin-is-not-the-future-of-securities-settlement/>

BITCOIN BAD, BLOCKCHAIN GOOD?

Bankers can be forgiven for being confused about blockchain technology. For months they've been told that blockchain and other distributed consensus technology can revolutionize the payments, clearing, and settlement infrastructure of the financial system, but that the bitcoin blockchain just won't do. This suits bankers fine, as few were ever anything but dismissive of the world's most popular cryptocurrency.

But then earlier this year, Nasdaq (2015) announced a project that will actually use the bitcoin blockchain to "facilitate the issuance, transfer, and management of private company securities" on their Nasdaq Private Market platform. So, what is going on?

No sooner had the press release circled around the small and sometimes befuddled group of financial types devoted (or at least instructed) to exploring this technology, when IBM's Richard Brown (2015) comes out with the warning to "ignore Bitcoin at your peril." And then this was followed by Chris Skinner's (2015) post, which suggests that the bankers' confusion is really just a case of the financiers having never understood the inner workings of bitcoin in the first place:

"So why would someone as intelligent and informed as Reid Hoffman – and Marc Andreessen, Richard Branson, Wence Cesares, Jon Matonis, et al – be so pro-bitcoin when the banks are not. My answer is that most of the people dissing bitcoin haven't looked under the hood.

So here are two test questions for all of you reading this and thinking Bitcoin Bad, Blockchain Good.

One, have you actually read Satoshi Nakamoto's white paper?
Two, can you explain to me exactly why the blockchain is good?
I don't do this, as I don't want to embarrass anyone, but I'm guessing that 99% of the Bitcoin Bad, Blockchain Good people would answer no to both questions."

Leaving the provocative aspect aside, he's probably right in that most bankers would answer "no" to his two questions. This might also be true for the majority of bitcoin's most vocal cheerleaders. Skinner then proceeds to the argumentum ad verecundiam and quotes the abstract of Nakamoto's (2008) famously elegant white-paper:

"A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures

provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers."

This article breaks down this abstract sentence by sentence and gives a non-technical explanation of how bitcoin works and why it's interesting. Then there will be a discussion to explain why the idea of using the bitcoin blockchain for securities settlements is misguided. This is not meant to knock Nasdaq for choosing a bitcoin meta-protocol for their project. It's a good way for them to cut their teeth in this area without devoting much capital expenditure. However, those who think that this news portends a future securities settlement architecture based on the bitcoin blockchain couldn't be more wrong.

A SIMPLE INTRODUCTION TO THE BITCOIN PROTOCOL

The first two sentences of Nakamoto's abstract make clear the design objectives behind bitcoin.

"A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending."
[Nakamoto (2008), 1]

It has been known for a long time, mainly because it is rather obvious, that cryptographic signatures and public keys can be chain-linked to form an unforgeable record of transactions for, say, digital cash (or any ledger record for that matter). Crypto proof replaces the notary. Counterfeiting ledger assets is impossible, and theft or misappropriation cannot happen without gaining access to the asset owner's private key [see appendix].

But you still need an authoritative record of these transactions somewhere, such as a database, or else there is no way to prevent someone from spending his or her digital cash more than once (a

“double-spend”). If one party gives as second a crypto-proof that some asset belongs to it and that it has been sent to the second party, the second party has no way of knowing that the first party hasn’t already done that with someone else, unless both parties can refer to a definitive ledger of timestamped and crypto-signed transactions, a ledger maintained as a database hosted by some trusted third-party, perhaps. The third-party cannot forge any ledger entries, so what is the problem with this setup? What are Nakamoto’s “main benefits” that are lost?

Firstly, there are two problems:

- The third party could delete a transaction, reversing history
- The third party could censor a transaction, i.e., refuse to enter it into the ledger.

And secondly, it’s not just the third party itself who has this power, it’s also the government who regulates them, or the hacker who infiltrates them. For Nakamoto (2008), using a trusted third party for this task loses some of the “main benefits” of the crypto setup because third parties have a real-world identity (a registered business, an IP address, etc.) and if known, these third parties could be censored by governments, hacked, or shutdown.

One of the key design goals behind bitcoin is censorship resistant digital cash. So, with this design goal in mind, how can a record of crypto-signed transactions that is both authoritative (in the sense that there is consensus on its veracity) and censorship resistant be created? Nakamoto provides a solution to this problem in the next part of the abstract:

“We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power.” [Nakamoto (2008), 1]

Bitcoin is a peer-to-peer network. It’s flat. It is architecturally decentralised. There’s no “bitcoin server” where the chain-linked blocks of transactions (transactions that are themselves also chain-linked via crypto signature) are centrally stored. Instead, the transaction record is stored redundantly by many nodes on the network. Anyone can be a node on the network anonymously – bitcoin is a “permissionless” network.

What does Nakamoto mean by “the network timestamps transactions”? Most people (especially people in financial markets)

understand a timestamp to mean something generated by an accurate clock. But this, remember, is a peer-to-peer network, so it doesn’t have a clock. The nodes on the network have clocks, but since these nodes could be anyone, the timestamp of any given node can’t really be trusted. So how exactly does the network “timestamp transactions”?

What Nakamoto means here by “timestamp” is something less precise: the ordering of the blocks of transactions, i.e., this block of transactions came immediately after that block of transactions. This is ordinal time, a relative timescale. It is in this sense that the “network timestamps transactions.” And how it does this is ingenious: “hashing them into an ongoing chain of hash-based proof-of-work.”

This is where many people get lost. The basics are actually rather simple, but it is important to understand some preliminary concepts first. A “hash-based proof-of-work” is a solution to a problem, a hash problem. The “hash” refers to a branch of mathematical functions called “cryptographic hash functions,” which have an interesting feature in that whatever data you put into one of these functions, they return a pseudo-random number of the same bit size. It’s not really possible to predict what the function will return given a certain input, without actually computing the function yourself. Between inputs and outputs, there is no pattern.

For example, here is the SHA256 hash (the same hash function used by the bitcoin protocol) of the input “Goldman Sachs”:

b0aad912e3a3d9c1be503c154c0580531709862a

Change that string by just one character and you get something entirely different: here is the hash of “Goldman Suchs”:

a0b9a202da83ea581e0306f28115b7c6e10c8483

In bitcoin, the hash problem is: “input into the hash function (1) a number of transactions along with (2) the hash of the previous block of transactions, and (3) an arbitrary number N; if the hash function returns a value below some number D (called “difficulty”, a varying number defined by the protocol), problem solved, if not, increment N and repeat.” There’s no way to solve this problem except through iteration (setting your computer to the task of running billions of hash computations until you solve the hash problem).

This is why it’s called “proof-of-work.” The problem was difficult to solve, it required the computer to do some work. But once it’s solved, you can prove to someone else that you did the work to solve it. Just show them the data (a number of transactions plus

the hash of the last block) and that winning number N , and they can calculate the hash. If the hash value is the same below- D number that you say it is, you've proved that you solved the problem. The problem is hard to solve, but the solution is easy for others to verify.

This is how the bitcoin network timestamps transactions. The nodes on the network ("miners") collect transactions that bitcoin senders broadcast and each works at solving the hash problem over a set of transactions. Whenever a node solves the hash problem, it broadcasts the block of transactions along with the proof-of-work. The other nodes verify the work and start hashing on top of that block (i.e., including its hash in the input of the hash problem).

And this is what Nakamoto means by "forming a record that cannot be changed without redoing the proof-of-work." Nodes on the network build on top of the "longest chain" of blocks. If an attacker wanted to reverse the history, say, five blocks back, he or she would have to redo the proof-of-work of those five blocks before other nodes would start accepting that his version of history is the correct version (because it's the longest chain). And that's no mean feat. As Nakamoto states:

"The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers." [Nakamoto (2008), 1]

This is a neat result. If every node follows the rule that the chain-linked set of blocks with the most work behind it is the blockchain, then every node's local copy of the blockchain will be exactly the same. And if an attacker wished to maliciously replace part of the "sequence of events witnessed" by the network (e.g., one where he made a big payment to someone) with an alternative version of history (e.g., one where he didn't make that payment), he would have to redo the latest work of the longest chain, and do this work at a faster rate than the rest of the network. Hence, he would need to control over 50% of the network's CPU power.²

And that, in a nutshell, is bitcoin's security guarantee. If you're comfortable believing that an attacker is unlikely to ever pull together more than half of the network's computing power, you can trust the veracity of the blockchain's record of transactions. Unlike with the case of a database hosted by a third party, there's no easy way for record entries to get "deleted" from the blockchain.

Here is a really important point to remember, however. All those hash problems that are being solved – the enormous amount of

computational power that is "securing the network," as it is popularly described – are not securing the network in the way that, for instance, a computer that encrypts a message secures its contents from prying eyes. There is no fancy math behind the security of bitcoin. The only reason that a cryptographic hash function is used is that a hash-based proof-of-work problem has the property of being "hard-to-solve-but-easy-to-verify." You need that asymmetry in solution/proof; the network would grind to a halt if everyone had to redo everyone else's work. That's why bitcoin miners aren't spending all that computational power on something useful like, for instance, genome sequencing. With most useful computations, you generally have to trust that the computer did them correctly, the computer can't prove to you that it computed correctly. But with a hash problem you can easily prove that you did the computational work to solve it, even though the solution is utterly useless math.

So, the security behind proof-of-work isn't "based on math" (as some misleadingly say). Those hash computations are there for one simple reason: to make it expensive to offer a block of transactions to the network that the other nodes on the network will accept as valid. This is an economic model of security, not a cryptographic one. Proof-of-work requires an attacker to make a substantial capital outlay (in computing power and electricity) to have any chance of pulling it off.

THE DESIGN GOAL: CENSORSHIP RESISTANT DIGITAL CASH

Nakamoto envisioned a distributed, shared ledger of transactions based on a principle of "one-CPU-one-vote" (although today you need dedicated sha256 hardware, so it's really more like a computing oligarchy than a computing democracy, a discussion point that is outside the remit of this paper). One might ask: Why not have a similar set up but instead use the principle of "one-node-one-vote," thus sidestepping the expensive and wasteful proof-of-work?

The answer to that question is the single most important idea to take away from the bitcoin protocol. One-node-one-vote works only if you have a way of authenticating the real-world identity of the node, because otherwise a single attacker could just masquerade

² In actual fact, researchers [Eyal and Sirer (2013)] have demonstrated that it is possible in theory to attack the bitcoin network with less than half of the network's computing power: the threshold is closer to 1/3 instead of 1/2.

as a number of different identities and gain control of the network, which can't tell whether 1000 nodes are really 1000 different people/entities or just one person behind them all, pulling the strings. This is called a Sybil attack³ in the computer science literature, and authenticating node identity is one way of mitigating that attack vector. However, Nakamoto settled on a more ingenious solution, the hash-based proof-of-work that was explained above.

Remember Nakamoto's design goal: the creation of "censorship-resistant" digital cash. Prior to bitcoin's popularity, privately created electronic money existed in a hostile political environment, to put it mildly.⁴ Authentication wasn't an option, because if the real identities of the nodes are known to all, governments could compel those nodes to censor transactions and apply KYC/AML procedures on the transaction senders, or, more extremely, just criminalise it and indict the operators behind the nodes. The "one-CPU-one-vote" idea behind hash-based proof-of-work is a solution that addresses the Sybil attack problem without relying on identity authentication. Instead of proving to the network that you're a unique flesh-and-blood person, you can prove to the network (without revealing your identity) that you've spent a lot of electricity and computing power, brute-forcing a solution to a meaningless math problem.

The bitcoin protocol is not only architecturally decentralised, it is also politically decentralised. The network has no gatekeepers, no permission is needed to join. The only admission criterion to contributing to the network's consensus is access to computational power.

"THE BITCOIN PROTOCOL IS NEITHER PERFECT NOR ANTI-FRAGILE"

At the beginning of this article, two main problems with using a ledger hosted by a trusted third party were pointed out:

- The third party could delete a transaction, reversing history
- The third party could censor a transaction, i.e., refuse to enter it into the ledger.

Nakamoto's hash-based proof-of-work beautifully solves the second problem. It is also designed to solve the first because bitcoin transactions are designed for irreversibility. And when bitcoin is cast in the role of distributed ledger platform for X (e.g., securities settlement), people are fond of describing the bitcoin blockchain as an "append-only distributed ledger for X."

But this is only a design goal, and because it is a design goal that has been subordinated to censorship resistance, the bitcoin protocol can provide no guarantees that this supposed "append-only" distributed ledger doesn't actually have a delete button accessible to an attacker who has a sufficient incentive and resources to attack the network and reverse blocks of transactions with impunity. Nakamoto (2008, 1) himself points this out in the abstract: "As long as a majority of CPU power is controlled by nodes that are not co-operating to attack the network, they'll generate the longest chain and outpace attackers." But if an attacker has access to more than 50% (actually, closer to 30%, see above) of the network's computing power, all bets are off.

In March of this year, Cornell University computer science professor, professor Emin Gün Sirer tweeted: "The #Bitcoin protocol is neither perfect nor anti-fragile. The main protecting force has been people's good will and lack of sophistication."

Emin is right. And this benign state of affairs is unlikely to persist if the bucket shops, which are today the only avenue for shorting bitcoins, are eventually replaced by professional derivatives markets. And it will certainly go away if billions of dollars worth of securities are represented through meta protocols on the bitcoin blockchain as some have eagerly extrapolated from the Nasdaq announcement. For then, attackers will have a way of constructing a scalable payoff for attacking the network: shorting the market in size. Acquiring a substantial portion of the network's hashing power is not an insurmountable goal. What is required is a sufficiently large monetary incentive to execute the attack. Putting billions of dollars worth of financial assets on the bitcoin blockchain materially changes an attacker's incentives.

Bitcoin transactions can be reversed if the attacker is willing to make the capital outlay to acquire the hardware and expertise and pay the electricity bill required to pull it off (bribing a few large mining pools is probably the path of least resistance). If the attacker is successful, the attack in theory costs nothing: the attacker collects the mining award of the blocks he solved, which "replace" the original transaction history, and which now is the chain with the most work behind it.

To the uninitiated, it might seem crazy that this ostensibly "append-only" distributed ledger that is the bitcoin blockchain contains

³ See https://en.wikipedia.org/wiki/Sybil_attack

⁴ See, for example, the Liberty Dollar case: https://en.wikipedia.org/wiki/Liberty_Dollar

an avenue for deleting history. After all, everyone saw those blocks of transactions before they were overtaken by the attacker's new blocks. Nobody will be fooled that the protocol's "network time-stamp" corresponds to the ordering of transactions that actually occurred. But that's how the protocol works: "the" bitcoin blockchain is the chain of blocks with the most work behind it. This is the price you pay for the censorship-resistant design.⁵

THE BITCOIN PROTOCOL AND SECURITIES SETTLEMENT

The idea of "coloring" nominal quantities of bitcoin to represent security interests and piggyback a distributed ledger of financial assets on top of a politically decentralised digital cash system is wrong. Having "looked under the hood" of the bitcoin protocol, it is possible to see why.

To serve as a replacement for the legacy technology implementing registered, book-entry assets, a distributed ledger of financial assets will have to ensure a tight correspondence between what the ledger and the law say is the state of who-owns-what. This is obviously incompatible with a protocol based on anonymous transaction validators: the law will not treat a ledger record as authoritative if everyone knows that the current longest chain contains transaction blocks generated by an anonymous attacker. But the bitcoin protocol has no mechanism for dealing with this scenario, no mechanism for bringing ledger state and legal state back into alignment. How could it? Remember Satoshi's design goal: "censorship-resistant" digital cash. The price paid for that goal was a proof-of-work consensus model where the chain with the most work behind it is "truth" as far as the protocol is concerned.

The financial system and its regulators go to great lengths to ensure that something called "settlement finality" takes place. There is a point in time in which a trade brings about the transfer of ownership – definitively. At some point, settlement instructions are irrevocable and transactions are irreversible. This is a core design principle of the financial system because ambiguity about settlement finality is a systemic risk. Imagine if the line items of financial institution's balance sheet were only probabilistic. You own X of Y with 97.5% probability. That is, effectively, what a proof-of-work based distributed ledger gives you. Except that you don't know what the probabilities are because the attack vectors are based not on provable results from computer science but economic models. Should a settlement system be built on that edifice?

Of course not. And fortunately, it doesn't have to be because there are many ways to design distributed, shared ledgers. And it is

unlikely that censorship-resistant securities transactions are the reason why financial institutions are looking at distributed consensus technology. Financial institutions' goals are rather different from Nakamoto's. Increased transparency is one, largely driven by the belief that regulators will grant concessions on capital charges for trades cleared through settlement systems that offer this. Efficiency through automating the back office is another. However, the main goal is probably increasing the speed of trade settlement.

From the experience of the author, this motivation perplexes many engineers, who understand well that distributed consensus technology is much slower than database technology. Proof-of-work protocols like bitcoin's are the slowest of the lot by far (and with only probabilistic ledger state to boot – censorship-resistance is expensive). But even far more efficient consensus algorithms will underperform the most basic relational database technology.

And yet it takes days to settle trades in book entry assets. This fact is only puzzling to those laboring under the mistaken assumption that custody accounting in the financial system is somehow centralised. It's not. Records are distributed throughout the system by thousands of different institutions, each one maintaining their own siloed accounts and constantly reconciling against each other to come to agreement on the global state of who-owns-what, or who-owes-what-to-whom. It is, in a sense, a form of distributed consensus: consensus-by-reconciliation. And consensus-by-reconciliation is very slow, expensive, and hard to automate. It is this technological infrastructure of consensus-by-reconciliation that the bankers, quite rightly, see being replaced by distributed, shared ledgers. This is a different problem from the one Nakamoto tried to solve, as a careful reading of Satoshi's abstract alone makes perfectly clear.

REGISTERED VERSUS BEARER ASSETS

Nothing in what has been discussed here is meant to take away from the inspired, brilliant solution that Nakamoto implemented for censorship-resistant digital cash. And, furthermore, that design goal is, in my opinion, a worthy one. Society should have digital cash that replicates the same anonymous and permissionless properties that are already enjoyed with physical currency.

⁵ When Nakamoto says that the longest chain "serves as proof of the sequence of events witnessed," I'm inclined to think he should have used the word "evidence" rather than "proof."

But a proof-of-work blockchain is only suitable as a distributed ledger for value that society is prepared to treat as a bearer asset. Physical cash is (almost) like this. A shop owner doesn't "due diligence" his customer to make sure that the \$10 note the customer is about to hand over rightfully belongs to him. In practice, when it comes to physical cash, possession-is-ownership.

The same principle applies to the bitcoin blockchain. Possession (of a private key) is ownership (at least in the anarchic, code-is-law jurisprudence of the bitcoin protocol), regardless of how one came into possession, for there is no way for the blockchain to discriminate among spend transactions of coins obtained through legitimate trade, defrauding a counterpart (e.g., via a double-spend), or theft of someone's private key.

But the proposition that security interests and other property titles should also be cast in the same bearer asset mould will go nowhere. Few actually want this, and, in any case, few jurisdictions will actually allow it. In fact, it's looking increasingly likely that few jurisdictions will even grant bitcoins bearer asset status.

The advocates of putting property titles on the bitcoin blockchain will likely object at this point. They will say that through meta protocols and multi-key signatures, third-party authentication of transaction parties can be built in, and we can create a registered asset system on top of bitcoin. This is true. But what's the point of doing it that way? In one fell swoop, a setup like that completely nullifies the censorship resistance offered by the bitcoin protocol, which is the whole *raison d'être* of proof-of-work in the first place. These designs create a centralised transaction censoring system that imports the enormous costs of a decentralised one built for censorship-resistance – the worst of both worlds.

If you are prepared to use trusted third parties for authentication of the counterparts to a transaction, then there is no compelling reason for not also requiring identity authentication of the transaction validators as well. By doing that, you can forego the gross inefficiencies of proof-of-work and instead use a consensus algorithm of the one-node-one-vote variety, which is not only thousands of times more efficient, but also places a governance structure over the validators that is far more resistant to attackers than proof-of-work can ever be.

REFERENCES

- Brown, R., 2015, "Blockchain is where banks have the most obvious opportunity. But you ignore Bitcoin at your peril," Richard Gendal Brown: Thought on the future of finance, blog. Available at: <http://gendal.me/2015/05/12/blockchain-is-where-banks-have-the-most-obvious-opportunity-but-you-ignore-bitcoin-at-your-peril/> and accessed August 15, 2015
- Eyal, I. and Sirer, E. G., 2013, Majority is not Enough: Bitcoin Mining is Vulnerable. Research Paper, Cornell University. Available at: <http://arxiv.org/pdf/1311.0243v5.pdf>
- Nakamoto, S., 2008, "Bitcoin: A Peer-to-Peer Electronic Cash System," white paper. Available at: <https://bitcoin.org/bitcoin.pdf>
- Nasdaq, 2015, "Nasdaq Launches Enterprise-Wide Blockchain Technology Initiative," press release, May 11. Available at: <http://ir.nasdaqomx.com/releasedetail.cfm?ReleaseID=912196> and accessed August 15, 2015
- Skinner, S., 2015, "Repeat after me: Bitcoin Bad, Blockchain Good," Financial Services Club, blog. Available at: <http://thefinanser.co.uk/fsclub/2015/05/repeat-after-me-bitcoin-bad-blockchain-good.html> and accessed August 15, 2015

APPENDIX

A colloquial illustration of the Diffie-Hellman-Merkle key exchange and Blockchain

We begin by illustrating how a private and public key can be used to exchange information in a secret way, to avoid man-in-the-middle attacks. We then comment on the differences with Bitcoin Blockchain.

Public Key exchange (PKE)

A has to send a message to B that must not be deciphered by anyone except B, in case the message falls in wrong hands (someone in the middle). A and B need to share a secret code to transform the message.

This secret code needs to be known to both of them and no one else.

A uses B's public key, say $K_{Bpublic}$ (accessible to anyone) and his own private key (secret, known only to A) $K_{Aprivate}$ to create a mixed key $KA1$.

$KA1$ = calculation based on $K_{Aprivate}$ and $K_{Bpublic}$.

Similarly, B uses A's public key $K_{Apublic}$ (accessible to anyone) and his own private (secret, known only to B) $K_{Bprivate}$ to compute

$KB1$ = calculation based on $K_{Bprivate}$ and $K_{Apublic}$.

$KA1$ is sent from A to B and $KB1$ from B to A. Anyone intercepting $KA1$ or $KB1$ and knowing $K_{Bpublic}$ and $K_{Apublic}$ wouldn't be able to get $K_{Aprivate}$ or $K_{Bprivate}$ by inverting the above calculations because this inversion is too hard computationally. In particular, B will not be able to compute $K_{Aprivate}$ and A will not be able to compute $K_{Bprivate}$.

Now both A and B do a second calculation using their respective secret keys, getting

B computes: $KA1B$ = calculation based on $KA1$ and $KB_{private}$.

A computes $KB1A$ = calculation based on $KB1$ and $KA_{private}$

By the nature of the operations and by relevant commutative properties, $KB1A$ and $KA1B$ are the same key. So now both A and B share a key $KB1A = KA1B$ that both can use to communicate secretly.

The information exchanged, namely $KA1$ and $KB1$, does not allow anyone intercepting it or the opposite party to find the secret codes of the sending party. The fact that only the mixed information $KA1$ and $KB1$ is exchanged and that separating the mixture is not possible in practice, due to computational difficulty, is at the basis of this public key encryption exchanges.

Public keys and Blockchain

In Blockchains one does not actually use PKE for encrypting (nothing on a blockchain is encrypted). One instead uses them for digital signatures. Suppose I encrypt some message M with my private key. Now anyone can decrypt that message with my public key (so privacy isn't our goal here, obviously). And by decrypting that message, anyone can prove that I wrote M , which is why we call that a crypto signature.

So the idea of using PKE to implement digital cash is elegantly simple. Everyone's public key acts like a sort of account number. So if I want to pay X digital cash to you, I create a message/transaction that says "I, the owner of this public key $K1$, pay public key $K2$ X bitcoin" and then I sign that message with my private key. Now the transaction ledger can verify via crypto proof that I'm really the person who controls the $K1$ "account" and nobody can tamper with my instruction (because that would break the signature) so there's a mathematical proof that $K1$ instructs to pay $K2$ X coins.

The implementation details are more complicated (but elegantly so), but this is really the essence of it. This one aspect of blockchain tech is over 20 yrs old and is really quite simple.

It's the distributed consensus over a ledger of such crypto-signed transactions that's more deep and difficult and where the real innovation lies.

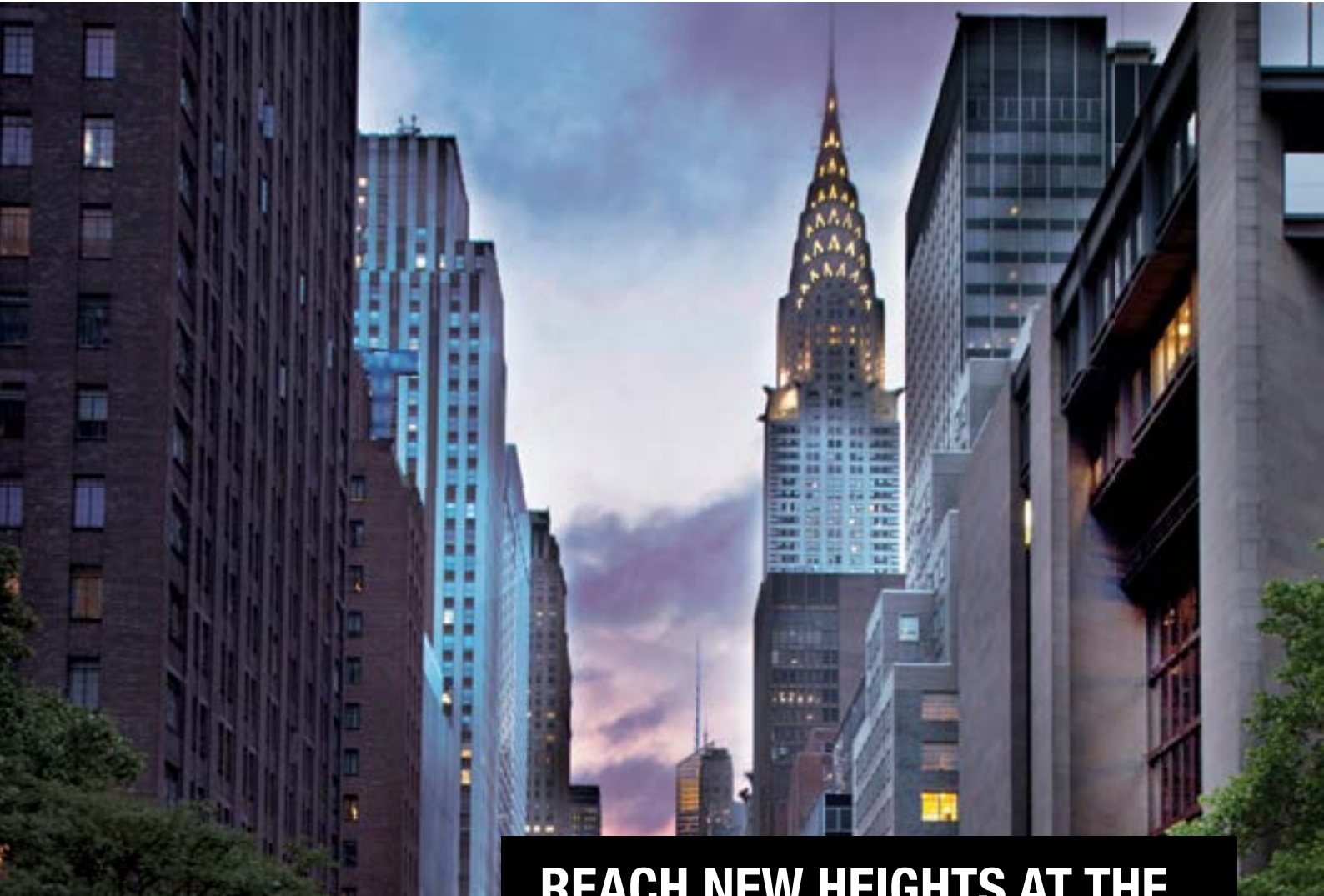
Layout, production and coordination: Cypres – Daniel Brandt, Kris Van de Vijver and Pieter Vereertbrugghen

Graphics: DuKemp

© 2015 The Capital Markets Company, N.V.

De Kleetlaan 6, B-1831 Machelen

All rights reserved. All product names, company names and registered trademarks in this document remain the property of their respective owners. The views expressed in The Journal of Financial Transformation are solely those of the authors. This journal may not be duplicated in any way without the express written consent of the publisher except in the form of brief excerpts or quotations for review purposes. Making copies of this journal or any portion thereof for any purpose other than your own is a violation of copyright law.



REACH NEW HEIGHTS AT THE ZICKLIN SCHOOL OF BUSINESS

TOP RANKED:

Top 100 Nationally, Top 3 in NYC, Top 5 in NYS for Graduate Business Programs —*US News and World Report*

#55 – Full-Time MBA Programs —*Forbes Magazine*

#58 – Part-Time MBA Programs —*US News and World Report*

#4 – Top Colleges for Finance and Financial Management —*USA Today*

#1 – “Best Bang for the Buck” in Northeast Region —*Washington Monthly*

www.baruch.cuny.edu/zicklin

BaruchCOLLEGE
ZICKLIN SCHOOL OF BUSINESS



CAPCO

**AMSTERDAM
BANGALORE
BETHESDA
BRATISLAVA
BRUSSELS
CHARLOTTE
CHICAGO
DÜSSELDORF
EDINBURGH
FRANKFURT
GENEVA
HONG KONG
JOHANNESBURG
KUALA LUMPUR
LONDON
NEW YORK
ORLANDO
PARIS
SAN FRANCISCO
SINGAPORE
TORONTO
VIENNA
WASHINGTON D.C.
ZÜRICH**

