Mine the gap: Bitcoin and the maintenance of trustlessness

Gili Vidan & Vili Lehdonvirta

Accepted for publication in *New Media & Society*

Abstract

Subscribing to a techno-utopian discourse which replaces institutions and experts with 'trust in code,' digital alternative currency Bitcoin is pitched as a 'math-based money' governed by incorruptible code rather than human regulators. In three case studies, which occurred between 2013-2015, we examine this system at moments of breakdown. In contrast to the discourse, we find that power is concentrated to critical sites and individuals who sometimes manage the system through ad-hoc negotiations, and who users must therefore implicitly trust—a contrast we call Bitcoin's 'promissory gap.' But even in the face of such contradictions between premise and reality, the discourse is maintained. We identify four authorizing strategies used in this work: conflating people with devices, assuming actors conform to notions of economic rationality, appealing to technical expertise, and explaining contradictions as temporary bugs. We contend that these strategies are mobilized widely to legitimize a variety of applications of algorithmic regulation and peer production projects.

Keywords: cryptocurrency; algorithmic regulation; Bitcoin; distributed ledger technology; critical code studies; peer production

Gili Vidan is a PhD candidate at the Department of the History of Science at Harvard and a fellow at the Science, Technology, and Society Program at the Harvard Kennedy School. Her work is broadly concerned with questions of governance at the intersection of digital information technologies, law, and politics. Her dissertation traces technical attempts to solve the problems of trust and transparency, especially through the development of electronic payment systems and public key cryptography in late 20th- and early 21st-century US. She holds a MSc in Social Science of the Internet from Oxford University and a BA from Harvard College. gvidan@g.harvard.edu

Vili Lehdonvirta (PhD, University of Turku) is an Associate Professor and Senior Research Fellow at the Oxford Internet Institute, University of Oxford, a Fellow of the Alan Turing Institute, and Hugh Price Fellow of Jesus College, Oxford. He is an economic sociologist researching how digital technologies shape the organization of economic activities in society, using conventional social research methods and novel data science approaches. His book *Virtual Economies: Design and Analysis* (with Edward Castronova) is published by MIT Press and translated to Chinese by China Renmin University Press. He is currently leading a major research project on online labour markets, funded by the European Research Council. vili.lehdonvirta@oii.ox.ac.uk

Introduction: Trusting code

'In Code We Trust' reads the cover of the May 2015 *New York Times Magazine* money issue dedicated to technological economic innovation. The image depicts a US dollar bill dissolving into colorful bits that appear halfway between confetti and pixels. Part of the issue is dedicated to Bitcoin, an online digital currency which first launched in 2009, promising its users an alternative to transactions mediated by banks and credit card companies. In online discussions about an appropriate motto for the new digital currency, 'code' had to compete with 'cryptography,' 'math,' and 'numbers' for being worthy of its users' trust. Perhaps true to the decentralized spirit of Bitcoin, all four versions are available for purchase in the form of shirts, mouse pads, and mugs. The chosen focal object, however, code, is apt; Maurer and colleagues (2013) argue that the value of Bitcoin as a form of money is rooted in the trust users place in its software code. Tim Wu reiterated the claim in a recent op-ed, arguing that finance is joining the trend, letting 'computer code take over' from humans (2017). Beyond Bitcoin, a huge variety of 'blockchain' applications and other platforms relying on algorithmic management are legitimated through this 'trust in code' discourse. In this article we explore how 'trust in code' is maintained despite the reality of it falling short of its promises.

In the turn of the 20th century, the social theories of Weber, Simmel, and Tönnies characterized 'modernity' as a shift away from the personal ties of traditional society towards state and other formal institutions as trusted mediators of impersonal economic relationships. Over a century later, contemporary social theorists point to the uneasiness resulting from such reliance on formal institutions and experts (Giddens, 1991; Beck, 1992; Jasanoff, 2016) and their associated 'technologies of distance' (Porter, 1995: ix). Distrust in banks and financial institutions in particular has given rise to 'alternative currency movements' that attempt to supplant mainstream money and financial institutions (North, 2007; Dodd, 2014). Since 2009, Bitcoin and other cryptocurrencies have entered public discourse as a prominent alternative currency movement.

Bitcoin and related digital currencies are referred to as 'cryptocurrencies' because of their reliance on cryptographic authentication technologies instead of personal ties or formal institutions as the means to verify transactions. Tracking how much money each

person has and making sure they only spend what belongs to them is carried out by a distributed network of computers running compatible cryptocurrency code, an arrangement that promised to render mediation through commercial banks unnecessary. The same code takes care of managing the total amount of currency in circulation, doing away with the need for a central bank. The movement has technological and ideological connections to 1980s and early 1990s US 'crypto anarchist' engineers and thinkers (May, 2001; Myers West, 2017; Swartz, 2018) and the technology builds on earlier academic research (Narayanan and Clark, 2017). But the movement's contemporary hero is Satoshi Nakamoto, a pseudonymous group or individual responsible for creating Bitcoin, the first and most popular cryptocurrency.[1] Nakamoto's expressed ideology is neither traditionalist nor classical modernist, but techno-utopian: he expects people to trust neither friends nor authorities, but program code. Through code, Nakamoto expects to reap the benefits of a modern impersonal economy without the cost of having to rely on a centralized power to regulate it. He follows a vision of trust in institutions and individuals not as something to be secured through technical means (Nissenbaum, 2001), but as a problem to be eliminated altogether, replaced with trust in the code itself.

What does it mean to trust code? When Bitcoin users replace the US dollar bill's invocation of 'God' with 'Code,' we argue that they attach a dual meaning to code as the object of their trust. The first is the notion of code as an impartial and incorruptible ruler and regulator of relationships. Unlike bankers and politicians, code is assumed to have no greed, fear, or ideology, and can be trusted to regulate the economy with a steady robotic hand. Tim O'Reilly (2013) describes the ideal government as one that embraces 'algorithmic regulation,' or regulation that automatically adjusts to received data. These are classical modernist ideals; Weber frequently likened the ideal bureaucracy to a 'technically rational machine' that provides fast, precise, predictable, and impartial regulation (Weber, 1978 [1922]: 811). Today's bureaucratic institutions—banks, governments—fall short of this ideal, but thanks to technological progress, goes the narrative, these institutions can finally be replaced with actual machines. This mechanical objectivity, in both its modernist conception as well as its current code-enabled iteration, is thus a struggle against particular forms of subjectivity (Porter, 1995: ix). Scholars of science and technology have long argued that human subjectivity always retains a

foothold or more in sociotechnical systems (MacKenzie, 2001). Technical fixes fail to supplant the need for trust, or the two turn out to be incommensurate (Nissenbaum, 2001). But rather than borrow from these established theorizations of trust and technology, we start from Nakamoto's and his followers' own imaginaries of 'trustlessness'—solving the problem of trusting individuals or centralized institutions—as the thing to be explained. In a series of breakdowns, Bitcoin reveals a persistent gap between this goal of trustlessness and its practical operations. By examining how this 'promissory gap' between promise and reality is bridged in the case of Bitcoin, we offer an account of the broader allure and staying power of imaginaries of algorithmic regulation.

The second meaning of 'trust in code' is to trust in the processes through which Bitcoin's code is produced. Here the narrative departs from classical modernist ideals and assumes an anarchist flavor. The legal code that is executed by state bureaucracy is produced by politicians, experts, and other power holders, whose legitimacy is questioned due to the perceived shortcomings of politics. As points of centralized power, these institutions are inherently susceptible to error or misconduct. In contrast, the code executed by the Bitcoin network is, in principle, produced by the community of Bitcoin users through processes of 'peer production.' Even when the code is subject to alterations, the narrative goes, members of the community may accept or reject these changes, thus reaffirming their trust in the (re)written constitution. In his study of open source software communities, Kelty defines such communities as 'recursive publics.' A recursive public is 'vitally concerned with the material and practical maintenance and modification of the technical, legal, practical, and conceptual means of its own existence as a public' (Kelty, 2008: 3). Kelty argues that the act of sharing source code is what makes these communities a well-defined recursive public, and creates a moral and technological order. Morally, it creates the ethos of open, free, and shareable information, and technologically, it allows for 'forking'—the emergence of alternative versions of a source code (Kelty, 2008: 119). But at critical moments, the Bitcoin network demonstrates the uneven dynamics of peer production: unaccountable centers of power, inability to reconcile conflicting interests, and consequently questionable ability to guarantee collectively-held values and just distribution.[2] We therefore seek to critically

evaluate what constitutes the 'trust in code' discourse and how it is maintained in the face of the promissory gap of algorithmic regulation and peer production. That code should be a trusted object is our point of analysis, rather than a finding.

In examining both meanings of Bitcoin's 'trust in code' discourse we seek to go beyond what Zeiwitz calls 'algorithmic drama,' which accords algorithms a mythic agency (2016: 5), and focus on the human work that upholds such myths. But the origin myth itself—Nakamoto's 2008 white paper—plays a critical role, as it sets out the promise of trustlessness that is being maintained through the discursive work of powerful actors in the network. We therefore begin our analysis with this fabled moment. Nakamoto aimed to create a system where people needed only to trust the code, the content of which they could themselves verify, as the everyday regulator of the economy; and in which the code would not be imposed by a central authority, but instead emerge from what he saw as a fundamentally democratic process. Following Susan Leigh Star's (1999) approach to the study of infrastructure, our analysis then traces the considerable frictions Bitcoin's advocates encountered in their attempts to realize Nakamoto's vision and solve 'the problem of trust' as the technology was expanding its reach from 2013 to 2015. We focus on some of the centralized sites of power in both the everyday social and material infrastructures of the Bitcoin economy as well as in the processes through which its code was produced and adopted into use. We explore how, even in the face of such contradictions between promise and reality, Bitcoin's political order continued to persist—how key actors mobilized a variety of authorizing discursive strategies to maintain 'trust in code.' The infrastructure of Bitcoin's trustlessness, we show, is as discursive as it is material. The making of the technical category of 'code' involves a lot of discursive work, by human actors, that make this infrastructure appear invisible. The same authorizing discursive moves—conflating people with devices, assuming subjects to be self-interested rational individuals, appealing to technical expertise, and explaining contradictions as temporary bugs—are also used in legitimating other projects where code and community supposedly transcend the messiness of politics. Through a study of Bitcoin's moments of breakdown and their resolution we therefore also present a wider critique of algorithmic regulation and the dynamics of software peer production.

Nakamoto's problem: Designing for trustlessness

Nakamoto's 2008 white paper, which provides a blueprint for the design of a new payment system, lays out what exactly his problems were with existing forms of digital payment. The document has cemented its status as a constitutive moment in the history of Bitcoin and has since been exhaustively analyzed for clues regarding Nakamoto's true identity. But it can also be read as an ideological blueprint for restructuring the role of trust in society.

For starters, the white paper emphasizes decentralization and the need to eliminate the trusted third parties that normally vouch for the exchanging parties' credibility and creditworthiness. Nakamoto's paper refers to this structure as the 'weaknesses of the trust based model' (2008: 1). What privileges central third parties such as banks is their ability to designate trust. Eliminating them from the transaction chain requires also eliminating the need for trust within a transaction. Trust, for Nakamoto, means assurance that a payment will not be reversed after a merchant has performed a service or delivered a good. While in physical transactions there is a way to confirm the finality of an exchange through cash, 'no mechanism exists to make payments over a communications channel without a trusted party' (2008: 1). The problem of trust is the problem of centralized authentication of transactions. Within a network of exchange, if one actor (e.g. a bank or a credit card company) holds monopoly over transaction authentication, that actor has gained power over the network's participants. Centralized trust, for Nakamoto, is a shorthand for centralized power.

To eliminate the centralization of power, Nakamoto's white paper puts forward a design for a 'peer-to-peer electronic cash system.' Each participant in the system runs a software on their computer that allows them to issue transactions to other participants in the network, but also defines what kinds of transactions are permissible. The software allows participants to spend only balances that they can prove they own by means of cryptographic keys. The software also regulates how new currency is issued, and places limits on the quantity of money in circulation. There is no central server or administrator to enforce these rules—the participants are regulated only by the software running on their computers.

To prevent participants from modifying their copy of the code so as to issue transactions that are against the rules, each transaction needs to be verified. Instead of having a single trusted party verify the transactions, the white paper describes an elaborate lottery-like system that chooses a verifier at random from among the network's participants. A new verifier is chosen approximately every ten minutes to handle a block of recent transactions. Power to alter the record of transactions is thus decentralized to the extent that no single participant holds any meaningful amount of it. The opportunity for abuse is negligibly small, because the probability of a participant being chosen to verify and re-verify their own transaction is infinitesimal.

Nakamoto's design seeks to assure the verifiers' integrity through a system of economic costs and incentives. To incentivize participation in the lottery and thus in the verification of transactions, the system rewards each chosen verifier with an amount of newly minted Bitcoins. But participating in the lottery is costly: it requires spending computing power and thus electricity to try out solutions, one by one, to an otherwise meaningless mathematical puzzle. Imposing this artificial cost on participants is necessary, because it makes it uneconomical for one party to acquire all the lottery tickets and become a central verifier controlling the system. This system of imposing a cost is known as 'proof-of-work', and those who bear the cost and participate in the lottery are known as 'miners.' The record of verified transactions they produce is a 'blockchain'.

The above rules are encoded in a part of the Bitcoin core software known as the 'protocol.' Participants are otherwise free to modify their copy of the software, but if they unilaterally change the protocol part, their copy may become incompatible with the rest of the network and miners may start to reject their transactions. The protocol and thus Bitcoin's rule set can only be changed if, roughly speaking, the majority of miners as measured by CPU power endorse the change. The purpose of the proof-of-work system thus extends beyond the day-to-day verification of transactions, and into the decentralized governance of the system's long-term trajectory: 'proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote' (Nakamoto, 2008: 3).

Nakamoto's consensus is one composed of 51% agreement among the nodes in the network, and the tethering of the representation of those nodes to proof-of-work has created a link between computing power and control. Nakamoto here seems to appreciate Actor-Network Theory's call for symmetry by equating participants and nodes, votes and CPUs. His imagined community fully embraces hybridity. Nakamoto concludes:

> The network is robust in its unstructured simplicity. Nodes work all at once with little coordination…They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism. (2008: 8)

We thus arrive at the heart of Bitcoin's decentralization—a community of CPUs. The power of authentication, previously centralized through trust in third parties, is now decentralized through its delegation to CPU power. Trustlessness has been established, based on the premise of distributed computational power. Previous studies of Bitcoin's materiality have revealed some of the shortcomings of Nakamoto's vision (Mallard, Méadel and Musiani, 2014; De Filippi and Loveluck, 2016). But they have stopped short of accounting for its discursive staying power. In the following section, we trace such moments of breakdown, and restoration of faith in its promise, to argue that Bitcoin and blockchain are as much discursive projects as they are material ones.

Breakdowns and bottlenecks: Sites of centralization

In the years since Nakamoto published his white paper and released the first version of the code, Bitcoin has grown into a huge network with hundreds of thousands of users, and reached new heights of valuation, with some economists and technologists warning of a bubble (Kharif and Leisig, 2018). This popularity and investment followed a tumultuous period of breakdowns and controversy, beginning in around 2013. Is Bitcoin's growth despite its difficulties a vindication of Nakamoto's trustless design? To obtain purchase on this questions, we follow Susan Leigh Star's call for the ethnographic study of infrastructure (1999). According to Star, one of the key characteristics of

infrastructure is its invisibility up to the point of breakdown, when its otherwise taken-for-granted components come under scrutiny. In Bitcoin, these breakdowns reveal centers of power in the ostensibly decentralized machinery of the cryptocurrency. In looking at infrastructural breakdowns, we are able to make visible the work that is inevitably part and parcel of the operations of a network such as Bitcoin, as well as the discursive work needed to make the infrastructure invisible in the first place. Our empirical accounts draw on news articles, discussion forum posts, and other public online sources, as well as published scholarship. Observing in each case the ways in which these breakdowns are resolved and 'trust in code' restored allows us to explain how Bitcoin continues to persist as a model of algorithmic regulation and peer production.

Bitcoin's financial gatekeepers

Nakamoto assumed that users would be running the Bitcoin software and storing their bitcoins on their own computers. In practice, as Bitcoin gained popularity, this became increasingly uncommon. Most users chose to use so-called exchanges and web wallets to store their bitcoins and issue transactions. These are convenient web-based services that save users the hassle of installing software and keeping it up to date. Even users who otherwise store their bitcoins themselves must briefly deposit coins with such a third party when buying or selling bitcoins for national currency via an exchange.

As long as exchanges and web wallets functioned smoothly, their gatekeeping role in the Bitcoin economy went relatively unexamined. But the fragility of the system was experienced by hundreds of thousands of users in early 2014, when Mt. Gox, at the time Bitcoin's largest exchange, collapsed. On February 7th, 2014, Mt. Gox disabled the withdrawals of bitcoins from users' accounts, meaning users who had stored their bitcoins in wallets hosted by the exchange could no longer access them (Hals, 2014). A couple of weeks later the exchange announced its intention to file for bankruptcy after experiencing technical issues which enabled the 'loss or theft' of 744,000 coins (Popper and Abrams, 2014). The event affected more than just Mt. Gox users, which themselves were a significant proportion of the Bitcoin community. It adversely affected the financial system as a whole. At the time the market value of bitcoin was about $820, but it fell to $528 by early March (http://blockchain.info/charts/market-price).

Initially, Mt. Gox placed the blame with the Bitcoin protocol, saying that a bug called 'transaction malleability' was at fault (Fuller, 2014). Bitcoin's core developers responded that, though transaction malleability was a known issue, it was the exchange's own implementation of its third-party software that caused the technical failure (Andresen, 2014a). The core developers sought to draw a distinction between Bitcoin's code, which remained trustworthy, and third-party code, which was external to the integrity of Bitcoin's design. If users chose to trust exchanges, the core developers argued, it was a choice they made at their own risk. The confinement of fault and attribution of untrustworthy code to Mt. Gox was cemented in August 2015 when the company's former CEO was arrested on charges of embezzlement. But Mt. Gox's fall was not an isolated incident: Moore and Christin (2013) reported that 18 out of 40 Bitcoin exchanges established during the preceding three years had closed, with customers' balances often wiped out.

Even without exhibiting such sudden breakage, exchanges use a variety of methods external to the Bitcoin network to verify the initial buy-in of new users, requesting state-issued documentation or proof of access to bank accounts. Negotiation of participants' trustworthiness is thus delegated to exchanges, which in turn delegate it to state and financial institutions. The delegation of trust has come full circle, back to its origins as Nakamoto's problem of centralization. But even as the promissory gap became apparent through exchanges' mediation of initial Bitcoin buy-in, the demarcation of code that is internal to the Bitcoin protocol and external developments such as the exchanges sustained the belief that Nakamoto's design was within reach as soon as such external interferences could be resolved. The delegation of trust was a temporary stopgap measure on the way to a truly trustless system.

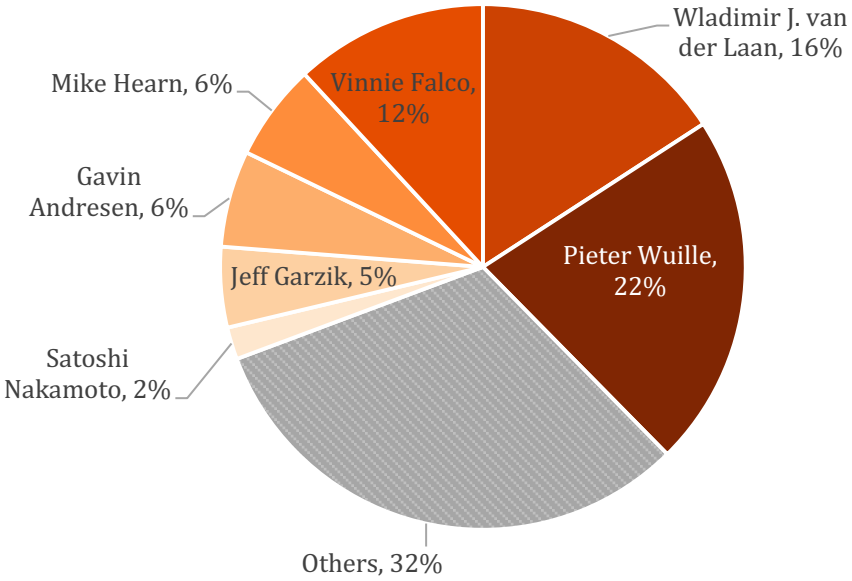
Bitcoin's de facto lawmakers

The transaction malleability bug draws our attention to another power structure under Bitcoin's hood that went relatively unquestioned until a breakdown became imminent. The Bitcoin protocol, despite its status as the constitution and the enforcer of the order, is still a piece of software—it is a set of documents written by humans for computers to interpret. It is a well-established tenet in software engineering that software is never

perfect: despite programmers' best efforts, defects or 'bugs' are almost guaranteed to remain. For this reason, software development is never completely finished: any software that is intended to be used on an ongoing basis must be maintained, that is, developers must remain on hand to release a patch or a new version each time a critical bug is discovered. Another reason why software needs to be maintained is that it interacts with other software and hardware, and whenever any of these are altered, it may have to be altered, too. Any live software system is not a static artefact, but an ongoing sociotechnical project.

Satoshi Nakamoto developed the initial version of the Bitcoin software, but soon ceased to maintain it. Around mid-2010, he reportedly handed over control of the project to Gavin Andresen, an Australian-born software engineer living in the United States. Andresen subsequently put in place an organizational model for Bitcoin software development and maintenance that retained its basic form for several years. It consisted of a core developer team, appointed by Andresen, who had write access to the official code repository. The core team was supported by volunteers, who were in principle free to propose changes to the code, but only changes approved by a core developer could enter the repository. Nakamoto also handed over to Andresen a special 'alert key' that allowed software update alerts to be issued to computers running the software. In other words, a group of then five white men comprising the core development team had full write access to Bitcoin's constitution, and no one but them. Figure 1 shows who had in practice written Bitcoin Core's code as of May 2015.

Figure 1: Bitcoin Core active lines of code by developer, 14 May 2015



Note: Data produced with *gitinspector* from the *bitcoin/bitcoin* master branch downloaded from Github on 14 May 2015

The core developers played down their powerful role by suggesting that they merely took care of the 'plumbing' (Dalais, 2015): applied bug fixes and carried out routine technical improvement work. But not all changes to the software were uncontroversial. In 2015, a controversy about the size of 'blocks' on which transaction are recorded started to draw attention to the core team's position. Due to an initial design choice in the Bitcoin protocol, the peak capacity of the Bitcoin network was limited to about seven transactions per second. For comparison, the Visa payment network processes around 2,000 transactions per second and has a peak capacity of around 56,000. If Bitcoin was to achieve mainstream adoption as a transaction platform, Andresen argued, changes had to be made to increase its capacity (Dalais, 2015). But such a change was thought to have significant implications to the nature of Bitcoin's financial system. Opponents claimed it would lead down a path where Bitcoin comes to resemble an interbank settlement network, where few people could access it directly, most having to transact through institutions like exchanges and web wallets. The 'block size debate' was an outstanding debate among developers since at least 2013 but reached a fever pitch in

mid-2015. The debate was at once ideological, technical, and commercial, as different Bitcoin startups and stakeholders stood to gain or lose depending on the path the system was set on.

The way in which the controversy unfolded illuminates the de facto political institutions and power holders behind Bitcoin's 'trusted code.' Whereas Nakamoto occupied to role of an unknown designer of a perpetual motion machine in Bitcoin lore, Andresen and the rest of the core development team emerged as powerful figures, but not as complete autocrats. For one, their power was limited by the 'Development Process,' a short set of rules outlined by Andresen in 2010, stating that significant changes required 'broad consensus' from the 'community,' not just among core developers. Many 'community members' or stakeholders deliberated the block size question, from developers and users to investors and academics, using mailing lists, forums, social media, and even mainstream media, but consensus was elusive. And as the controversy gathered steam, the legitimacy of the core team members' privileged position came under question. Users on the main Bitcoin discussion forums started such threads as 'Looking for evidence to support theory of Gavin [Andersen] successor to Satoshi' and 'Please list arguments against the idea of taking away Gavins' alert keys.' In 2016 Andersen was finally ousted and his privileged access was revoked. The block size debate both restructured the core development team itself, and drew the Bitcoin community's attention on the role such individuals played in the process of trusting code.

Code's allure as at once a virtual process and a material object (what Wendy Chun calls 'the conversion of event into location', 2011: 53) played a key role in bridging over this significant promissory gap. When code revealed itself to be written by specific people, the network as a whole, represented as a collective of physical CPUs, acted as checks on centralized power. Another tangible limitation on the core team's power, brought up by the core team itself as well as by its opponents, had to do with the process of mining. Roughly speaking, any update to the Bitcoin software had to be installed by miners representing more than half of the network's computing power for any protocol changes contained in the update to be effective. In Nakamoto's vision, this functioned as a referendum on any proposed changes, where users 'vote with their CPU power' (Nakamoto, 2008: 8). But this embodied metaphor obfuscates trends in CPU

accumulation that animated much of Bitcoin's movement from the ideological project of hobbyists and into a professional industry.

Bitcoin's computing power blocs

Mining is a fundamental design feature that allows Bitcoin to function as a seemingly decentralized system. By contributing CPU power towards solving arbitrary puzzles, miners participate in the day-to-day enforcement of the system's rules but also 'vote with their CPU power' to ratify or turn down proposed changes to those rules. The reward from successfully mining a single block is significant, but as the total amount of computing power contributed to the network increased, ordinary users could hope to win in this lottery only extremely rarely. At the same time, mining involves constant costs. A miner's success rate depends on the amount of CPU power they are able to contribute in relation to other miners, leading miners into a competitive cycle of hardware investments. The cost of supplying electricity to such specialized hardware is also very significant.

To make their income flows more regular, miners came up with the idea of pooling their resources and distributing the rewards between members in relation to the computing power contributed. Slush, pseudonymous creator of the first mining pool established at the end of 2010, explained his motivations:

> I created the Bitcoin pool service with the rising difficulty of mining in mind because I'd like to return my investment to mining hardware. There used to be days when I never found a block at all, even with the strongest GPU on the market, which made me uncomfortable. The periodic micropayments from the mining pool offer a steadier payout, which lowers the riskiness of my investment. (2014)

Some pools have operated like mutual organizations, some are managed as businesses that take a cut from the miners' rewards, and some are simply companies that own mining hardware and perhaps are better described in Lana Swartz's terms as 'factories' (2018). These pools have become part of Bitcoin's infrastructure, a mechanism working in the background to keep transactions flowing, requiring little

attention from users who were not directly involved in mining. That changed when the growth of mining pools reached a breaking point in early 2014.

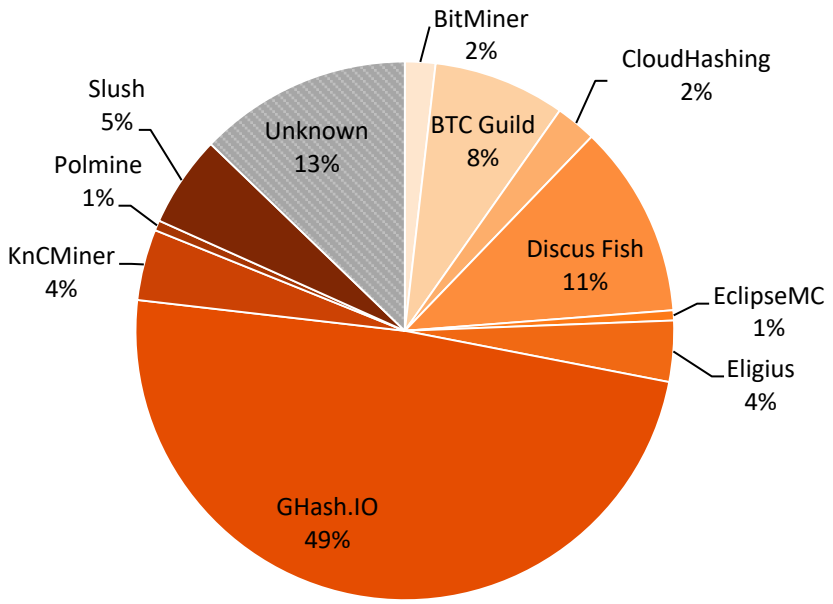
Near the end of 2013 some posters on Bitcoin forums began to voice concern over GHash.IO's increasing share of computing power in the Bitcoin network. The largest pool at the time, GHash.IO's homepage displayed a banner claiming it was 'trusted by over 180,000 miners.' On January 9th, 2014 news outlets reported that GHash.IO was approaching 45 percent control of Bitcoin's computing power (Wile, 2014), perilously close to the 51 percent point that constitutes control over the network. In face of criticism and alarm from users, GHash.IO issued a public statement, claiming that, although it sees its increasing share as a generally desirable thing, it will 'take all necessary precautions to prevent reaching 51% of all hashing power, in order to maintain stability of the bitcoin network' (GHash.IO, 2014). The pool went on to reassure the Bitcoin community that it 'does not have any intentions to execute a 51% attack.' Simultaneously, after a few hours, the pool's share dropped to 38 percent, likely because some miners within the pool decided to move to smaller pools for the sake of reassuring those concerned (Mims, 2014).

Malicious intentions or not, the January 2014 rise of GHash.IO constituted a breakdown in Bitcoin's trustlessness narrative. The protocol could not have prevented the pool from reaching 51 percent, nor could it have deterred manipulation of the blockchain from such a vantage point. The continual functioning of the protocol hinged on GHash.IO's promises not to exploit its powerful position. Bitcoin users found themselves relying on a third party who remained obscure and anonymous to them. The fact that Bitcoin continued to function after this brief controversy suggests that the pool's homepage was perhaps due for an update—between January and June 2014, GHash.IO could have claimed to be trusted not only by a certain number of miners, but by the entire Bitcoin ecosystem.

Despite GHash.IO's promises to take 'all necessary precautions,' a few months later the pool made waves again. This time GHash.IO's share did, for a few hours, reached the critical point of 51 percent (Hern, 2014). For the 24 hours period surrounding those critical moments on 13 June 2014, the pool's share averaged on 49 percent. Figure

2 shows an analysis of new blocks mined and added to the block chain during a period of 24 hours that day.

Figure 2: Distribution of Bitcoin Block Mining on 13 June 2014



Note: Data produced using *Block Explorer* API, linking to its identified origin each block mined and added to the blockchain, from block #305502 to block #305665.

This occurrence was more widely covered in the media (*The Economist*, 2014; Hern, 2014; Brustein, 2014; Casey and Vigna, 2014) and reaction from concerned users also seemed to be stronger. One prominent user announced that he was liquidating half of his bitcoins (Todd, 2014). GHash.IO issued another statement a couple of days after the 51 percent threshold made headlines, saying that the pool, 'never [has] and never will participate in 51% attacks or double spend against Bitcoin' (*Pressat*, 2014). Recognizing, however, that a reiteration of their intentions from a few months back was not sufficient, GHash.IO suggested creating a forum where leading mining pools and the core developers could negotiate the decentralization of mining—essentially a semi-formal institution for negotiating the distribution of power among an elite.

Core developers also responded to the incident. Gavin Andresen repeated his past assurance that potential attacks by a majority miner are highly unlikely since they are

contrary to the pool's economic incentive. Yet, despite this reassuring stance, Andersen still urged miners to turn to smaller pools rather than GHash.IO (2014b). Though aiming at inspiring confidence in the resilience of the Bitcoin protocol and network, what these institutional responses actually achieved was to bring into the foreground the social and political infrastructures that supported the technical arrangements. For the Bitcoin order to survive this moment of breakdown, a new social and political narrative had to be added to the 'trust in code' discourse: mining pools were rational, credible, and benevolent; the core developers were technically capable (the authority on whether risks are likely to be realized) and politically responsible (encouraging miners to turn resources to smaller pools). The original idea of a distributed community of CPUs was all but dead. The number of individual miners was counted in the thousands, while mining pools were estimated to command the CPU power of hundreds of thousands of people (Parker, 2015). Throughout these moments of breakdown, Bitcoin, both the network and the ideological commitment to 'trust in code', has been able to survive, thanks to the material as well as discursive interventions of its power holders.

Discussion: Strategies of maintaining trust in code

'Making the fantastic seem credible is hard work,' writes Lauren Buekes of her Science Fiction novel, *Zoo City*, a world in which the ability to text and email with spirits is as mundane as the ability to send money electronically (2010: 311). The moments of breakdown in Bitcoin's promised fantasy were bridged through the mobilization of a variety of discursive strategies that play on slippages between the meanings of code, work, and decision-making. Code seems to enjoy a unique epistemic status within the algorithmic regulation discourse—it is presented as transparent and predictable. Galloway has claimed that as an executable set of instructions, 'code is the first language that actually does what it says' (2004: 165–166). The division between tangible, knowable core code, considered as internal to the Bitcoin ecosystem, and the disruptive actions of actors outside of what constitutes the core allows for the network to maintain its narrative of algorithmic decentralization when facing contradictory evidence. Why do users continue to trust in code, especially in this particular code, in the face of such breakdowns?

The ‘trust in code’ discourse stands on two epistemic legs: algorithmic regulation, or the belief that automated processes of decision-making are less fallible than human institutions; and peer production, or a set of assumptions about the openness and inclusiveness of the code development process. Analyzing responses to the moments of breakdown explored in this paper, we can identify four authorizing discursive strategies mobilized by various actors in the network to maintain the narrative in times of breakdown. While not all three are necessarily held by the same actors at all times, and some can be read as contradictory, we argue that the concurrent existence of all four within the ‘trust in code’ discourse is the source of its staying power.

The first of these authorizing strategies is the collapse of users and their representations on the network into the aggregation of CPUs that power the network. This ambiguity with regards to the identity of the Bitcoin community—individual human actors or their dedicated machines—allows the network to be portrayed as a self-regulating system not susceptible to human foibles, and simultaneously as an enabler of direct action. Under the edict of ‘one-CPU-one-vote,’ any incident within the Bitcoin network is at once the expected result of running the protocol and the enforcement of an expressed consensus of its users. The Bitcoin protocol reimagines its constituency as a mass of CPUs.

A second authorizing strategy is borrowed directly from the toolkit of market liberalism—the assumption of rational, self-interested agents. When CPUs as stand-ins for a mass of individual users appeared to have been accumulated at the hands of a single actor such as GHash.IO, both the pool operators and the core developers issued statements to reassure users that it would not be in the pool’s rational self-interest to undermine the network or to go over the 51 percent threshold. Both of these strategies implicitly assume a situation in which the terms of action on the network are fixed. The Bitcoin network, much like the liberal market, is considered ontologically prior to the actions of actors that constitute this network. Developers did not hesitate to predict what the pools would or would not do based on rational choice, even though the original developer had failed to predict the (rational) emergence of pools in the first place.

The third authorizing discursive strategy we came across is the appeal to technical expertise. Indeed, a commitment to a technocratic order seems to underwrite much of the

previous two strategies. The belief that cryptographic know-how should grant particular actors in the network governing power enables the simultaneous elevation of Nakamoto's paper as the ultimate authority of keeping the network within the bounds of its intended purpose, and the acceptance of the Core Development Team as the legitimate body to carry out updates of the code. A commitment to technocratic order first enrolls users in the network through the promise of a decentralized system that ensures the need to trust no-one, and then, when the system's unsettled and unpredictable nature becomes visible, the technocratic order privileges certain actors as legitimate holders of centralized power until the infrastructure can be stabilized again. Collapsing the difference between users and CPUs further facilitates this technocratic structure, because if the Bitcoin network is composed of machines, then who better to rule it than engineers.

The fourth and final discursive strategy is casting problems as temporary bugs that will not be present in the final, ideal version of the code. Instead of critically reflecting on the shortcomings of the 'trust in code' narrative, participants are asked to ignore contradictions as limitations of a particular implementation of the code. Sites of centralized power are cast not as inherent consequences of the architecture, as features of it, but rather as temporary shortcomings to be overcome in later iterations of the code, as bugs to be patched. Bitcoin's and blockchain's initial appeal comes from the promise of a one-time buy-in into infallible code, which will be from that moment on fixed, knowable, and autonomous. Once issues of centralization emerge, however, the code then becomes a malleable experiment, subject to iterations and improvements to address the temporary aberration. If anything is maintained as fixed, it is the belief that trustlessness can be engineered, the belief that Nakamoto's elegant vision is almost within reach through minor technical adjustments. Participants are asked to trust if not this version of the code, then the next one, in perpetuity. It is of no consequence whether solutions are in sight today, because the peer production model will keep iterating until they are.

Implications to scholarship: Discourse made durable

One way to view the reemergence of centers of power within the Bitcoin network is as exemplifying Latour's assertion that 'technology is society made durable' (1990). The Bitcoin protocol, in this view, is fixing certain values and a power distribution that

privileges certain actors such as the financial gatekeepers, the technical lawmakers, and the hardware monopolists the analysis above explored. But as we suggested, what is puzzling about Bitcoin is not the fact that it recreates centers of power within its network, but rather that this gap between promise and reality, when made apparent, can still be overcome to uphold 'trust in code'. In addressing this question, we rather followed Leo Marx, in trying to see how technology came to be seen as autonomous, erasing the work, both technical and discursive, of politics and governance (2010 [1997]). We showed that the infrastructure of Bitcoin's trustlessness is as discursive as it is material, and that trustworthy code lives on not as an extant technology, but as a commitment. The same discursive strategies used to protect this commitment in Bitcoin can also be used to account for the broader allure and staying power of imaginaries of algorithmic regulation reflected in works such as O'Reilly (2013) and Wu (2017).

What distinguishes Bitcoin and other blockchain-based developments from past forms of technocratic governance, however, is the discursive role played by their characterization as decentralized technological systems. Past modes of bureaucratic, and increasingly machine-based, governance have likewise attempted to project a dichotomy between the fallibility of human actors and the smooth, predictable operations of machines. In the financial sector in particular, the introduction of computers has followed perceived weaknesses in the work of human actors (Kennedy, 2017). But in the blockchain mode of trustlessness, decentralization through a collapse of users and their CPUs further removes human agency from the visible operations of the system. There is no banker, bureaucrat, or even software developer at which the finger could easily be pointed. Code, as the trusted object of blockchain dreams, is thus even further removed from its conditions of production and maintenance than in the prior entanglements between social institutions and technological systems that Leo Marx (2010) was describing. Sensing this shift in how technology and institutions operate under 'trust in code,' the CEO of Intercontinental Exchange Inc., owner of the New York Stock Exchange, commented on cryptocurrencies: "[p]eople are more comfortable [with] technology than the institutions of government and society that I grew up with" (Vaghela, 2018).

All this is not to suggest that attempts to use digital technologies to facilitate more progressive forms of political order are necessarily doomed to fail. The implication is rather that such projects must make conscious efforts to develop political institutions that reach towards their professed ideals. We are critical of techno-utopian thinking that affords 'communities' or 'peer production' some form of inherent moral superiority independent of how they are in practice maintained. As the fantastical future commands the resources of the present, it is necessary to develop a heightened awareness of the power imbalances and epistemic commitments that are being projected forward.

To contemporary social theorists, Bitcoin could be used to exemplify the late-modern rejection of formal institutions and experts (Giddens, 1991; Beck, 1992). Instead of trusting formal institutions and institutionally recognized experts, people reflexively choose their own authorities to believe in, whether religious, political, or in this case, technological. According to Giddens (1991), this distrust is simply a continuation of the Englightenment era commandment to question all authority, initially directed towards autocracy and theocracy, but lately gaining vigor in Western societies to call into question even democratic institutions and the rules of everyday civil life. But it is the emergence of new and perhaps undertheorized sites of trust and power that must command further scrutiny (Jasanoff, 2003). Code, as a material and discursive object, for all its cultural slippages and ontological instability, is now taking hold of visions of how social ties are managed in the future, in ways that are proving surprisingly durable.

Acknowledgements

The authors wish to thank Luciano Floridi, Sheila Jasanoff, Alex Csiszar, Frank Hangler, and Jonathan Levin for their helpful comments on early versions of this research and to Dylan Mulvin, Sam Weiss Evans, and Susanne Freidberg for their input on the manuscript.

Notes

1. While Nakamoto's identity remains unknown, the chosen pseudonym is a Japanese male name, so we will refer to him using the male singular third person pronoun.
2. It is significant that Bitcoin's aim to function as a payment system relies on notions of what it means to be a member of a peer-to-peer payment system, and speaks back to the broader sociology of money (Maurer et al., 2013). Lana Swartz has outlined how the competing economic imaginaries of Bitcoin, premised on its potential functionality as a payment system, concern the very role of money in society (2018). But even such competing imaginaries still rely an initial valuation of code as a trustworthy object. Nigel Dodd argued that Bitcoin's ideology is in fact antithetical to its ability to function effectively as a form of money (2017). The analysis of whether Bitcoin fulfils the functions of prior electronic payment systems or the ontologically slippery category of 'money,' must therefore still attend to the question of trust placed in code.

References

Andresen G (2014a) Contrary to Mt. Gox's Statement, Bitcoin Is Not at Fault. In: Bitcoin Foundation. Available at: http://bitcoinfoundation.org/2014/02/10/contrary-to-mt-goxs-statement-bitcoin-is-not-at-fault (accessed 30 July 2014).

Andresen G (2014b) Centralized Mining. In: Bitcoin Foundation. Available at: http://bitcoinfoundation.org/2014/02/10/contrary-to-mt-goxs-statement-bitcoin-is-not-at-fault (accessed 30 July 2014).

Beck U (1992) *Risk Society: Towards a New Modernity.* London: Sage.

Brustein J (2014) Is One Bitcoin Miner Controlling the Entire System? *Bloomberg Businessweek*, 16 June. Available at:

http://www.businessweek.com/articles/2014-06-16/could-bitcoin-miner-ghash-dot-io-control-the-entire-bitcoin-system (accessed 30 July 2014).

Buekes L (2010) *Zoo City*. Nottingham: Angry Robot.

Casey MJ and Vigna P (2014). BitBeat: Mining Pool Rejects Short-Term Fixes to Avert '51% Attack.' *Wall Street Journal*, 16 June. Available at: blogs.wjs.com/moneybeat/2014/06/16/bitbeat-a-51-attack-what-is-it-and-could-it-happen (accessed 30 July 2014).

Chun WHK (2011) *Programmed Visions: Software and Memory*. Cambridge, Mass: MIT Press.

Dalais J (2015). Gavin Andresen: We Have a Bitcoin Soft Fork Going On. *CryptoCoinsNews*, 22 April. Available at: https://www.cryptocoinsnews.com/gavin-andresen-bitcoin-soft-fork-going-eta-increased-block-size-6-months-year/ (accessed on 18 February 2018).

De Filippi P and Loveluck B (2016). The invisible politics of Bitcoin: Governance crisis of a decentralized infrastructure. *Internet Policy Review* 5(3).

Dodd N (2014) *The Social Life of Money.* Princeton: Princeton University Press.

Dodd N (2017) The Social Life of Bitcoin. *Theory, Culture & Society*. Epub ahead of print 17 December 2017. DOI: 10.1177/0263276417746464

Fuller C (2014) MtGox Blames Bitcoin for Withdrawal Suspension, Core Developers Say Otherwise: Who Is Really at Fault? *International Business Times*, 11 February. Available at: http://www.ibtimes.com/mtgox-blames-bitcoin-withdrawal-suspension-core-developers-say-otherwise-who-really-fault-1554512 (accessed on: 30 July 2014).

Galloway A R (2004) *Protocol: How Control Exists after Decentralization*. Cambridge, Mass: MIT Press.

GHash.IO (2014) Press Release: Bitcoin Mining Pool GHash.IO Is Preventing Accumulation of 51% of All Hashing Power. Available at: http://ghash.io/ghshio_press_release.pdf (accessed on 30 July 2014).

Giddens A (1991) *Modernity and Self-Identity: Self and Society in the Late Modern Age.* Cambridge: Polity.

Hals T (2014) Mt. Gox Files U.S. Bankruptcy, Opponents Call It a Ruse. *Reuters*, 10 March. Available at: http://www.reuters.com/article/2014/03/10/us-bitcoin-mtgox-bankruptcy-idUSBREA290WU20140310 (accessed on 30 July 2014).

Hern A (2014) Bitcoin Currency Could Have Been Destroyed by '51%' Attack. *The Guardian*, June 16. Available at: http://www.theguardian.com/technology/2014/jun/16/bitcoin-currency-destroyed-51-attack-ghash-io (accessed on 30 July 2014).

Jasanoff S (2003) In a Constitutional Moment: Science and Social Order at the Millennium. In Joerges B and Nowotny H (eds) *Social Studies of Science and Technology: Looking Back, Ahead*. Springer, pp.155–80.

Jasanoff S (2016) Subjects of reason: goods, markets and competing imaginaries of global governance. *London Review of International Law* 4(3): 361–391.

Kennedy D (2017) The machine in the market: Computers and the infrastructure of price at the New York Stock Exchange, 1965-1975. *Social Studies of Science* 47(6): 888–917.

Kelty CM (2008) *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press.

Kharif O and Leising M (2018) All About Bitcoin, Blockchain, and their Crypto World: QuickTake. *Bloomberg*, December 11. Available on: https://www.washingtonpost.com/business/all-about-bitcoin-blockchain-and-their-crypto-world-quicktake/2017/12/11/ba566a26-de9b-11e7-b2e9-8c636f076c76_story.html (accessed on: 11 February 2018).

Latour B (1990) Technology Is Society Made Durable. *The Sociological Review* 38(S1): 103–131.

MacKenzie, D (2001) *Mechanizing proof: computing, risk, and trust*. Cambridge, MA: MIT Press.

Mallard A, Méadel C and Musiani F (2014) The Paradoxes of Distributed Trust: Peer-to-Peer Architecture and User Confidence in Bitcoin. *Journal of Peer Production* 4.

Marx L (2010 [1997]) Technology: The Emergence of a Hazardous Concept. *Technology and Culture* 51(3): 561–577.

Maurer B, Nelms TC and Swartz L (2013) 'When Perhaps the Real Problem Is Money Itself!': The Practical Materiality of Bitcoin. *Social Semiotics* 23(2): 261–277.

May T (2001) Crypto Anarchy and Virtual Communities. In Ludlow P (ed) *Crypto Anarchy, Cyberstates, and Pirate Utopias*. Cambridge, Mass: MIT Press, pp. 65–79.

Mims C (2014) The Existential Threat to Bitcoin Its Boosters Said Was Impossible Is Now at Hand. *Quartz*, 9 January. Available at: http://qz.com/165273/the-existential-threat-to-bitcoin-its-boosters-said-was-impossible-is-now-at-hand/ (accessed on: 30 July 2014).

Moore T and Christin N (2013) Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk. In Sadeghi A-R (ed) *Financial Cryptography and Data Security*. Berlin: Springer, pp. 25–33.

Myers West S (2017) Survival of the Cryptic. *Limn* 8. Available at: https://limn.it/survival-of-the-cryptic/ (accessed 11 February 2018).

Nakamoto S (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. Available at: https://bitcoin.org/bitcoin.pdf (accessed on: 11 February 2018).

Narayanan A and Clark J (2017) Bitcoin's Academic Pedigree: The concept of cryptocurrencies is built form forgotten ideas in research literature. *ACM Queue* 15(4): 1–30.

Nissenbaum H (2001) Securing Trust Online: Wisdom or Oxymoron? *BUL Rev.* 81: 635.

North P (2007) *Money and Liberation: The Micropolitics of Alternative Currency Movements*. Minneapolis: University of Minnesota Press.

O'Reilly T (2013) Open Data and Algorithmic Regulation. In Goldstein B and Dyson L (eds) *Beyond Transparency: Open Data and the Future of Civic Innovation*. San Francisco: Code for America Press, pp.289–300.

Parker L (2015) Number of Bitcoin Miners Far Higher Than Popular Estimates. *Brave New Coin*, 13 May. Available at: http://bravenewcoin.com/news/number-of-bitcoin-miners-far-higher-than-popular-estimates/ (accessed on 2 September 2015).

Popper N and Abrams R (2014) Apparent Theft at Mt. Gox Shakes Bitcoin World. *The New York Times*, 25 February. Available at:

http://www.nytimes.com/2014/02/25/business/apparent-theft-at-mt-gox-shakes-bitcoin-world.html (accessed on 30 July 2014).

Porter TM (1995) *Trust in Numbers: The Pursuit of Objectivity in Science and Public Life*. Princeton, NJ: Princeton University Press.

Pressat (2014) GHash.IO Is Open for Discussion, June 16. Available at: http://www.pressat.co.uk/releases/ghashio-is-open-for-discussion-93ee9eeb66b80e94bbe31705d451780e/ (accessed on 30 July 2014)

Slush (2014) Is It Safe? *BitcoinCZ Mining Aka Slush's Pool*. Available at: http://mining.bitcoin.cz (accessed on 30 July 2014).

Star SL (1999) The Ethnography of Infrastructure. *American Behavioral Scientist* 43(3): 377–391.

Swartz L (2018) What was Bitcoin, what will it be? The techno-economic imaginaries of a new money technology. *Cultural Studies*: 1–28.

The Economist (2014) Losing to Win, 23 June. Available at: http://www.economist.com/blogs/schumpeter/2014/06/bitcoin (accessed on 30 July 2014).

Todd P (2014) Why I Just Sold 50% of My Bitcoins: GHash.IO. *Reddit*. Available at: http://www.reddit.com/r/bitcoin/comments/281ftd/why_i_just_sold_50_of_my_bitcoins_ghashio (accessed on: 30 July 2014).

Vaghela V (2018) ICE Chief Says Crypto Trading Is a 'Trend We Can't Ignore.' *Bloomberg*, 9 April. Available at: https://www.bloomberg.com/news/articles/2018-04-09/ice-s-chief-says-crypto-trading-is-a-trend-we-can-t-ignore (accessed on 30 April 2018).

Weber M (1978 [1922]) *Economy and Society: An Outline of Interpretive Sociology.* Berkeley: University of California Press.

Wile R (2014) A Group of Miners Has Exposed One of Bitcoin's Fatal Flaws. *Business Insider*, 10 January. Available at: http://www.businessinsider.com/bitcoin-miners-approach-dangers-threshold-2014-1 (accessed on 30 July 2014).

Wu T (2017) The Bitcoin Boom: In Code We Trust. *The New York Times*, 18 December. Available at: https://www.nytimes.com/2017/12/18/opinion/bitcoin-boom-technology-trust.html (accessed on 11 February 2018).

Ziewitz M (2016) Governing Algorithms: Myth, Mess, and Methods. *Science, Technology & Human Values* 41(1): 3–16.