INSTITUTE OF COMPUTING
STATE UNIVERSITY OF CAMPINAS

Avenida Albert Einstein, 1251 – Barão Geraldo
13083-852 Campinas, SP, Brazil

INSTITUTO DE COMPUTAÇÃO

Campinas, February 22, 2018

To: Editor, NIST Reports
Ref: NISTIR 8202 (DRAFT): Blockchain Technology Overview

Dear Editor:

I am a Professor of Computer Science at the State University of Campinas (UNICAMP), in Campinas, Brazil, with Ph. D. from Stanford University (1989). Like almost all other computer scientists, I am very skeptical of cryptocurrencies and of "blockchain technology". However, unlike those colleagues (who, after noticing the fundamental flaws of the idea, generally pay no further attention to it), I have been closely following the phenomenon since 2014 – motivated chiefly by curiosity about its "sociology" and "ecology," but without disregarding its technical aspects.

Thus, considering my apparently very rare position, I feel obliged to share my views about the NIST draft report 8202. Unfortunately, it is extremely negative.

First, the technical description of blockchains in the draft is rilled with gross errors and misleading statements, and omits many important details. For example, when describing the Bitcoin blockchain, the authors incorrectly state (on line 607) that the hash of each block is stored inside the block but outside of the header, apparently unaware of why the Merkle root is included in the latter. They also state (on line 592, and again on line 905) that the transactions received by a miner are kept in an "unspent transaction pool;" thus conflating two distinct and important data structures, the *unconfirmed* transaction pool and the unspent *output* set. The latter error is like conflating the crankshaft and the suspension when describing how a car works, and talking about its "suspension crankshaft."

These errors inexcusable, since there are plenty of other sources that describe the the bitcoin protocol and blockchain much more clearly, in more detail, and more accurately than this draft does [1, 3]. Even the Wikipedia articles on bicoin, although biased in their evaluation, are better than this account. It is disconcerting to see the NIST, with its reputation for accuracy and technical leadership, would consider publishing such a shoddy work by amateur authors who do not even understand the basics of the Bitcoin blockchain, much less the technologies that it is supposed to replace.

Second, the organization and focus of the draft are quite confusing and inappropriate. The authors waste an excessive amount of space describing details of the Bitcoin blockchain and protocol that are not pertinent even to other cryptocurrency blockchain. Likewise, they discusses many aspects

Jorge Stolfi
Full Professor
http://www.ic.unicamp.br/~stolfi

stolfi@ic.unicamp.br
Tel: +55 (19) 3521-5858, 9-9639-5181
Fax: +55 (19) 3521-5847

of cryptocurrency blockchains without regard to whether they matter for non-currency ones; and aspects of permissionless blockchains as if they were valid also for permissioned ones.

Third, and worst of all, the authors are extremely biased in their evaluation of the virtues and potential of blockchain technology. They incorrectly state — without any supporting arguments or evidence — that it is better than traditional centralized database technology in many aspects, when in fact it is worse in *all* measurable aspects. The *only* supposed advantage of a blockchain-based ledger is that it would be "decentralized" — that is, it would not be operated by a central authority. It should be said "supposed" because, in fact, the technology does not even achieve that goal.

The draft reads like the prospectus of any of the myriad "blockchain" projects that have sprung up in the last few years, ostensibly to "disrupt" this or that activity. It is not surprisingly that none of those projects have yet produced any blockchain-based system that is effectively better than the centralized systems that it would replace. The cryptocurrencies themselves have failed miserably at their goal: none, not even Bitcoin, is an usable payment system, and none, not even Ethereum, have been useful for anything except financial scams.

The draft completely omits this most important fact. It also fails to cite any of the many of the publications that have been critical of "blockchain technology", such as David Gerard's comprehensive review of the technology and past projects [4]. In this regard, the draft reads like a primer on nutrition written by Herbalife distributors.

In my view, publication of the report as is (or maybe in any form) is likely to do more harm than good to the computer industry, since it is likely to convince companies and developers in all fields to waste time looking for "blockchain" solutions to their problems, that almost certainly will not serve their needs.

In the remainder of this document, please find detailed discussion of the points above.

Sincerely,

Jorge Stolfi

1 General remarks about blockchain technology

1.1 The only possible advantage of blockchain is decentralization

Basically, a blockchain is the worst possible ledger data structure one could imagine, in every aspect, including security, availability, and resistance to tampering. Satoshi chose it for bitcoin because his goal was to build a totally decentralized payment system — and that was the only structure that could be maintained in a totally decentralized way, by a swarm of uncoordinated, unregistered, unsupervised anonymous volunteer "miners".

Any structure more efficient than a blockchain would require some central control, if only to provide the locking functionality needed to implement atomic updates by multiple independent partners.

Incidentally, Satoshi himself chose a more efficient data structure — a balanced Merkle tree — to organize the transactions inside each block. As explained in section 2 of this review (about line 534 of the draft), that structure reduces the amount of information that a lightweight user needs to download in order to verify that a specified transaction was included in the blockchain. Satoshi could use that structure inside each block, because, while the blockchain as a whole is built in a decentralized fashion, the construction of each block is "centralized" — done by a single miner, independently of all other miners or users.

1.2 Permissioned blockchains make no sense

A permissioned blockchain, by definition, mush have some mechanism to "hire" or "fire" the miners — persons or entities that are allowed to append new blocks to the chain. That mechanism would have to be a centralized entity: either a single person or company, or a consortium of known entities bound by contracts. That entity must be trusted by all participants, since it indirectly controls the blockchain.

However, if a company or project has such trustworthy central management, then it should not waste time looking at blockchain technology. That central authority could directly manage the system, using traditional replicated database and digital signature technology. That way it could get **all** the alleged features of blockchain, but with advantages in all respects – including cost, speed, flexibility, and security.

1.3 Decentralization is a costly feature still looking for a need

I am not aware of any application where full decentralization is an overriding requirement, and for which the management of critical data could be entrusted to a swarm of unknown miners — that cannot even be prosecuted if they fail to do the expected job, or mess it up.

Not even Bitcoin is an example of such an application. For Satoshi, decentralization was not a need, but the goal itself: he basically wanted to solve a technical challenge ("design a fully decentralized e-payment system") that had been open for 25 years, and had even been proved impossible to solve. While he made an attempt to justify the need for decentralization as a way to reduce costs, bypass payment blockades, and avoid inflation, experience has shown that Bitcoin does *not* achieve any

of those things. That is, the blockchain solution is inadequate even for the application of internet payments, for which it was devised.

The distinguishing feature of blockchains — decentralization — is not a technical advantage per se. It does not make the system faster, more economical, more reliable, more flexible, or anything else. On the contrary, it has negative impact in all those aspects, especially speed and cost.

Decentralization could be only a political advantage, not a technical one. Still, it is hard to imagine an application where that feature would be desirable — and so desirable as to justify the inefficiency and all other disadvantages of a blockchain-based solution.

1.4 A permissionless blockchain must be a cryptocurrency

A permissionless blockchain, by definition, cannot have any mechanism to authorize or veto the miners who maintain it. It cannot even require the identification of miners, because any reliable ID mechanism would require a trusted central authority, such as the national government or some other clearing agency, that miners would be required to physically visit.

Obviously, one cannot hope that a swarm of unknown volunteer miners will properly maintain one's precious ledger data just out of good will. In order to argue that his protocol worked, Satoshi first had to provide an economic incentive to miners, in the form of a reward to be assigned to miners who succeeded including their blocks in the final (longest) chain.

Satoshi could not have specified that the reward would be paid in any national currency (such as dollars) or physical commodity (such as gold), since these options would require a "bank" of some sort that would keep custody of those assets and issue the payments. That bank would then be a central authority with the power to punish any miner who acted against its wishes.

Therefore, to be truly decentralized, a blockchain needs to have its own tokens to reward the miners; and those tokens must become valuable, and the miners must be able to exchange those tokens, and the transfers of tokens must be recorded in the blockchain. In other words, a truly permissionless blockchain must be a cryptocurrency blockchain.

1.5 Blockchains are logs, not databases

Blockchains are generally presented by enthusiasts as alternatives to the main databases used by centralized systems, such as the one that records the current balances of all accounts in a bank.

Those proponents fail to notice, however, that a blockchain does not directly record the *current state* of the system, such as account balances or land ownership. It is an *historical log* of all events that changed that status. A historical log must be, almost by definition, append-only.

Now, any well-engineered system should keep, in addition to modifiable databases that record the current state, also a full historical log — for auditing and statistical purposes. Even Satoshi's bitcoin mining software had, besides the historical log (the blockchain) a separate live database (the unspent output set).

Therefore, projects that may consider using a blockchain must be aware that it will only be a replacement for its historical log — not for its active database.

1.6 Blockchains do not ensure data avaliability

Contrary to claims of many suporters, the use of a blockchain does not ensure, per se, that the past history of the system will always be available for checking.

Large blockchains, such as those of cryptocurrencies, are not expected to be replicated in every node, since most users would not be able or willing to assume that burden. In the Bitcoin system, for example, most users are expected to be lightweight nodes, which keep at most the headers of the blocks, not the full blocks. The so-called "full nodes" are *supposed* to keep a complete copy of the Bitcoin blockchain; however, since those non-mining nodes have no incentives to do so, they cannot be trusted to actually do it.

In typical cryptocurrency systems derived from Bitcoin, the mining nodes have an incentive to validate the whole blockchain, and to send to their peers any new blocks as they are mined, by them or by other miners. However, a miner does not have to store the whole blockchain forever. He only needs to store the set of unspent outputs that can still be used as inputs, since that is all the data he needs in order to validate future transactions. While the protocol allows anyone to query any miner for past blocks, the miners are not obliged to fulfill such requests.

In general, most users of a large blockchain-based system would not keep the whole chain, and would to trust that "someone" is doing so for them. That is not as likely as blockchain enthusiasts claim. For example, the Ripple blockchain is missing approximately 32 thousand blocks at its beginning — because no one felt necessary to save them at the time [6].

The continuing availability of the full blockchain data ultimately depends on the existence (which is *not* guaranteed by blockchain technology) of *mirror sites* that will provide the contents of the historical log on request. The permanence is best assured if these mirror sites are numerous, physically and administratively independent, competently run, and are motivated to provide the service — if possible, by legal contracts. Well, this is exactly how traditional centralized database systems ensure permanence of their data; and they generally do it much better than cryptocurrencies or other systems based on permissionless blockchains — especially on the "competence" and "motivation" aspects.

1.7 Data in a blockchain is not immutable

It is often claimed that information recorded in a blockchain is permanent, or very hard to change. However, that is not true.

The simplest way to change a record in a blockchain is to enter another record that supersedes it. (The authors note this fact in line 988, yet often claim the opposite, e. g. in lines 933 and 1242). While the blockchain would still retain the original record and show the change, that is true also of traditional logs maintained by responsible systems, like banks and credit card processors.

The conditions and cost of such changes are not determined by the blockchain itself, but by the software that is used to interpret and manage it. The current Bitcoin software, for example, requires that any reversal transaction be signed with the private key of the address that received the original transaction — a property often stated as "bitcoin transactions are irreversible." However, other cryptocurrencies, or non-currency blockchains, may allow reversals and corrections in other

circumstances – such as when signed by the original sender, within a set grace period; or signed with some "master key". And even the Bitcoin protocol may be modified in the future to allow such operations.

The information stored blockchain can also be modified by changing the rules of the protocol, so that past records are interpreted in a different way. In an extreme case, the new rules might define special handling for specific past records. For example, the bitcoin protocol could be extended with the rule "the transaction with ID X shall be ignored" (thus effectively reversing it) or "the coins in address Y should be considered to be in address Z" (thus effectively modifying past transactions that paid to Y).

In the case of a permissioned blockchain, the central authority can force the miners to accept such changes. That may seem more difficult in a permissionless blockchain, but it is not possible to give sufficient guarantee that it will not happen.

Moreover, even a set of "rogue" miners with a minority of the hashpower can modify a permissionless blockchain by forking it: that is, by creating a new branch that starts before the block that is to be changed. While the majority of the miners will ignore that branch and keep working on the original one, the rogue minority might achieve its goals if it can convince some users that their branch is the only legitimate one. They could do that by preventing those users from receiving the majority chain, or by persuading them to use software that rejects it.

Finally, another way to change data in a blockchain is the rewind-and-rewrite process described in section 1.8 below.

1.8 Blockchains themselves are not immutable

While proponents admit that new records can supersede previous records, they generally claim that the blockchains themselves are "immutable," in the sense that past records cannot be deleted, modified, or interpolated.

However, that is patently not true. By definition, any "tail" segment of the blockchain *will* be discarded and replaced, if someone publishes another valid branch, sprouting from some previous block, that is longer than the current one, even if by a single block. Then every user will be bound, by the "longest chain" rule, to forget the previous chain and accept the new one instead.

Thus, in the case of a permissioned blockchain, any number of blocks at the end of the chain can be discarded and replaced by other blocks, if the parties that maintain it agree to do so. Since "permissioned" effectively implies the existence of an entity (even if a consortium) with power to authorize and exclude miners, that entity effectively has the power to impose such "rewind-and-rewrite" operations whenever it wants.

A permissionless and truly distributed blockchain with proof-of-work (PoW) can be rewound and rewritten by any hostile entity that has can set up more mining power than all the current miners combined; or if a majority of the miners can be convinced that it is in their interest to do so. (A similar statement could be made about proof-of-stake blockchains.) The authors themselves note this possibility in line 1200 of the draft; yet they still claim, in the very first sentence, that blockchains are "immutable.'

The cost of an "external" rewind-and-rewrite operation (by a hostile entity) is no bigger than the cost that was paid to create the segment to be rewritten. That cost will be prohibitive only if the blockchain itself is prohibitively expensive to operate. Namely, if a permissionless blockchain costs 100'000 dollars per day to operate, then data up to 10 days old can be *permanently* modified by a malicious entity at the *one-time* cost of a little over 1 million dollars.

An "internal" rewind-and-rewrite operation (by the usual miners) requires only obtaining the agreement of a majority of them (counted by hashpower). Such agreement is hardly impossible to achieve.

Indeed, the Bitcoin blockchain itself has seen three such "rewind-and-rewrite" events: in 2010 [9], 2013[10], and 2015 [11]. In each of those events, a bug in the software used by most miners caused the blockchain to become "invalid" — not according to that software, but according to the expectations of users and developers. In each event, a majority of the miners (counted by their total mining power) was convinced by developers to cancel a significant number of blocks the end of the current blockchain (51, 24, and 6 blocks, respectively), and build a new replacement branch, using a patched version of the software.

In those occasions, any valid transactions that were confirmed in the original chain were supposed to be confirmed again by the miners in the new branch. However, in one of those occasions, one payment was replaced by a different transaction, that sent the same coins to a different address, thus causing a $10,000 loss to a payment processor. That incident is categorical empirical evidence that even the massive mining industry of Bitcoin cannot ensure the immutability of the blockchain.

After a few years of fruitless discussion of that problem, Bitcoin enthusiasts have tacitly agreed to ignore it (just as we all became desensitized to the risk of a global nuclear war). However, any company or government that is considering the use of its own permissionless blockchain should be aware that a malicious party can sabotage its database, by spending, for only a few days or weeks, a little more than what that company will spend continuously to maintain it.

1.9 Permissionless blockchains are fundamentally flawed

Moreover, permissionless blockchain technology is fatally flawed, and cannot in fact provide a truly decentralized secure database.

In order to claim that his protocol worked, Satoshi had to make a critical assumption: that the majority of the miners (counted by their "hashpower", i.e. the computing power that they devoted to mining) would be "selfish greedy:" that is, they would only care to maximize their chance of adding the next block to what would become the final blockchain, thus claiming the corresponding reward. He also assumed that that those miners too would make the same assumption about the majority of their peers.

With these assumptions, he was able to prove that the blockchain was reliable, in a probabilistic sense. Specifically, he could claim and prove an upper bound to the probability that a transaction could be reversed after being confirmed — that is, that a block that was included in the longest branch of the blockchain would be later discarded, and replaced by some other block that might not contain that transaction. The bound decreased very quickly as more blocks were appended to that branch. With those assumptions, he concluded that all users (as well as those "selfish greedy"

miners) should always trust the longest known branch of the chain; and that any block in that branch that was sufficiently removed from its current tip. (In the case of bitcoin, a block followed by five or more blocks could be considered definitive for all practical purposes.)

Satoshi's assumptions above were essential to his proof. If the assumptions were violated – that is, a majority of the miners chose to pursue other goals than maximize their immediate expected gain — then no probability bounds could be proved about the permanence of the longest chain. An "unselfish" or "non-greedy" majority could could discard and re-create any number of blocks at the end of the chain.

Satoshi's fundamental assumptions seemed plausible at the time, since it was expected that hash-power would be distributed among thousands of independent miners who would have no reason to identify themselves. In that scenario, it would be very difficult for a single entity to force or motivate a majority of the miners to act in any "unselfish" way. It would be difficult also for a majority of the miners to collude and pursue a "non-greedy" strategy that, while not optimal for the next block, would give them greater rewards in the longer run. Then, the best strategy for an independent selfish miner would be the "selfish greedy" one expected by Satoshi.

What Satoshi and other early analysts failed to realize was that a large miner would have many advantages, and hence be more profitable, than two independent miners half its size. Thus, once the block reward was sufficient to motivate people to mine bitcoin, it also motivated the miners to merge and associate under a few large pools. Those factors inevitably caused bitcoin mining to become centralized into a handful of large pools. As I write, the three largest pools have 55% of the total hashpower in the world, and the six largest have 70%. Moreover, most of the pools and their operators are not anonymous, and more than 70% of the total hashpower is held by pools located in a single country (China).

In this scenario, Satoshi's essential assumptions cannot be taken for granted. If a handful of the large pools, comprising a majority of the hashpower, could chose to pursue some "unselfish" or "non-greedy" strategy — by bribes, threats, or the expectation of higher profits in the long-range — then one could no longer trust Bitcoin payments, even after hundreds of confirmations. In that case, Satoshi's proofs break down: it becomes impossible to conclude anything about the probability of certain "undesirable" events, like reversal of long-confirmed transactions. In fact, the current Bitcoin system cannot be considered decentralized: every user must trust that those few mining pools will continue to follow the "selfish greedy" strategy.

Indeed, it was only thanks to the concentration of Bitcoin mining that the system could be saved in the three "rewind and rewrite" incidents of 2010, 2013, and 2015. They developers were able to contact the major pools and convince them to be "unselfish" and "not greedy:" forgo the rewards that they had already collected in the buggy chain, stop trying to extend it, and instead work on the new branch, even though it was not the longer one. Those three events show conclusively that Satoshi's probability bounds are not valid in reality.

2 Specific remarks

Below are remarks about specific points of the draft report:

- Lines 113,265: *Blockchains are immutable digital ledger systems* [. . .] *no transaction can be changed once published*

  The first sentence in the report is already seriously wrong. As discussed in section 1.8 of this review, blockchain are not "immutable" at all.

- Line 120: *These currency blockchain systems are novel in that they store value, not just information*

  A cryptocurrency blockchain does not "store value". It only records the assignment of abstract balances to "addresses," and changes in those assignments. The attribution of value to those balances is provided entirely by the users, through marketplaces where national currencies, goods, and services are exchanged for access to those balances.

  In that respect, a blockchain is not innovative at all. Any digital financial ledger, such as the database of a bank or of a stock exchange, "stores value" in the same sense that a blockchain does. (The very first example of writing that survived from ancient Mesopotamia, more than 6000 years ago, were records of ownership and transfers of value.)

  In fact, the mapping of entries in a bank's ledger to the national currency is defined by laws and contracts, and the value of said currency is stabilized by its national Central Bank. In contrast, the blockchain of a cryptocurrency has no legal standing, and the currency itself has no stabilizing entity. Thus, the value "stored" in it is defined only by the fickle mood of its users and speculative traders. That is one reason why the value of all cryptocurrencies is so volatile.

- Lines 121,450: *a digital wallet—an electronic device (or software) that allows an individual to make electronic transactions*

  To be precise, a (cryptocurrency) wallet is a digital file that records a collection of (supposedly) unspent coin amounts in the blockchain, and the private keys that would allow spending them. What the report calls "wallet" is more properly called a "wallet management tool" – either a software that runs in some common computing device, such as a workstation or smartphone, or a special-purpose hardware and software combination.

  While the tool is often called just "wallet" in informal contexts, the full name should be used to avoid ambiguity. For one thing, a cryptocurrency wallet (file) can be used with different wallet management tools, possibly with trivial format conversions.

- Line 122: *wallets are used to sign transactions sent from one wallet to another*

  This is inaccurate. In all major cryptocurrency systems in use today, transactions do not record transfers between wallets, but only between blockchain *addresses*. Wallets, and the assignment of addresses to wallets, are not public information. The concept of "wallet" is not even part of the protocol.

  In fact, the same unspent coins may be recorded in several wallets, which may be accessible to different people. (Needless to say, only one of these people will be able to move those coins). And

transactions can be signed — even when the inputs are not recorded in any wallet — by software or hardware that is not a wallet management tool.

- Line 129: *Clarke's statement is a perfect representation for the emerging use cases for blockchain technology*

  I find Clarke's quote rather inappropriate here. Blockchain technology is actually *less* advanced than the technology used every day by many banks and credit cards, and it has still to enable any effects that would seem magical. (Unless one views as "magic" the unexpected and permanent disappearance of one's money through theft of private keys, collapse of currency exchanges, or wild price swings without any apparent reason.)

- Line 147: *on a blockchain, it is much more difficult to change data*

  As discussed in section 1.7 of this review, changing data in a blockchain is no more difficult than changing data in a traditional database.

- Line 147: *on a blockchain, it is much more difficult to* [...] *update the 'database' software* [...] *changes to the blockchain software may cause forking of the blockchain*

  On the contrary, updating "blockchain" software is not particularly difficult. It is in fact often easier than updating the software of massively used high-performance applications, like Visa or Google.

  A permissioned blockchain must have a specific process to decide changes to the protocol or to its interpretation and handling. The parties will then be bound by contract to update their software accordingly.

  Changing the protocol of a permissionless blockchain requires, first, convincing a majority of the miners to accept the changes. That has been done many times in the past in all cryptocurrencies, including bitcoin. Once a sufficiently large majority of the miners have agreed on upgrading the software and implementing the change at a given date, the remaining miners and users are practically forced to do the same.

- Line 163: *most blockchains use some common core concepts. Each transaction involves one or more addresses and a recording of what happened, and it is digitally signed*

  The notion of "addresses" is specific to cryptocurrency blockchains. Blockchains that are not intended for payment systems do not need to have addresses, and not all transactions need to be signed. (Even the bitcoin blockchain has one unsigned transaction in each block – the "coinbase" transaction, that assigns the block's reward to the miner.)

- Line 166: *a new hash is created for the current block's header to be recorded within the block data itself as well as within the next block*

  This is incorrect. The hash of the entire block's contents, minus the header, **is** included in the block header: it is the root of the Merkle tree. The hash of the header is *not* stored in the block, but only in the header of the next block.

- Line 170: *Each technology used in a blockchain system takes existing, proven concepts and merges them together in a way that can address problems that were previously difficult*

  That statement is quite misleading. *Everything* that a blockchain does could be (and was) being done previously, with much simpler, more efficient, and more reliable tools. The *only* novel thing that the blockchain brought is not a *what*, but a *how*: it would (in theory) allow some things to be done without a central authority. However, as argued in section 1.3 of this review, that is not a technical advantage; quite the opposite.

- Line 179: *blockchain technology is an important concept that will be a basis for many new solutions*

  That is not a technical fact. At this point, it is purely a statement of faith, that goes against logic and experience.

- Line 270: *It has enabled the success of e-commerce systems such as Bitcoin, Ethereum, Ripple, and Litecoin*

  Only Bitcoin has seem some limited use in e-commerce. Ripple was ostensibly designed to be a settlement system for banks (but has not been adopted by them). Ethereum was created to be a platform for smart contracts.

  None of these cryptocurrencies, not even Bitcoin, can be considered a successful payment system for e-commerce. Even the most ardent supporters of Bitcoin have been forced to accept this fact, and are redefining its goal as being a "store of value" rather than a payment system.

  If anything, those cryptocurrencies have exposed the fundamental flaws of the permissionless (non-permissioned) blockchain concept.

- Line 294: *Many electronic cash schemes existed prior to Bitcoin, but none of them achieved widespread use*

  This statement is unclear, since the authors do not define "electronic cash". By some definitions, it would be false: credit and debit cards have been successfully replacing physical cash well before 2008, and PayPal went public in 2002.

- Line 299: [The blockchain] *also enabled the issuance of new currency in a fair fashion to those users (sometimes called* miners *or* minters*) maintaining the blockchain*

  It is questionable whether the issuance of coins to miners was "fair." Satoshi himself mined about one million bitcoins, which is about 5% of all the bitcoins that (in theory) will ever exist. In fact, the mining reward mechanism was one of the factors that led to the concentration of most bitcoin mining into half a dozen pools.

- Line 300: *among other factors,* [Bitcoin's issuance process] *enabled lower transaction costs for using the system*

  The use of blockchain technology and the mining reward mechanism resulted in bitcoin being a ridiculously expensive payment system. Today, miners are paid more than 60 USD for each transaction that they confirm — more even than international bank wire fees for ordinary users.

What the coin issuance did was to push that cost on unwary bitcoin "investors," rather than the actual users. As I write, bitcoin "investors" are paying 18 million USD every day to the miners who operate the Bitcoin system — who process less transactions in a day than Visa processes in 10 seconds. Surely, Bitcoin must be the most inefficient electronic payment system ever devised by humankind.

- Line 305: *the blockchain enabled users to be pseudonymous*

  The pseudonymity of bitcoin is not a consequence of using a blockchain, and was not even a goal of the Bitcoin project. It was rather an unavoidable side effect of decentralization — because recording and certifying the identity of address owners would require some central authority, at least at national level, with control over the miners.

  Pseudonymity has been a curse for Bitcoin, since it resulted in illegal payments being its major use by far. It also helps enormously the theft of bitcoins by hackers, since it makes it almost impossible to identify the thief. I cannot imagine an application — not even electronic payments — where one would want transactions to be submitted by unidentified users.

- Line 310: [the Bitcoin blockchain] *greatly reduces the ability for users to double spend (sending the same digital asset to more than one user)*

  The blockchain in fact was the device that allowed Satoshi to (theoretically) *eliminate* double-spending, not just "greatly reduce" it. Any electronic payment system must prevent double-spending, and the developers of such systems have been doing it since computers were first used in banking. What the blockchain enabled (theoretically) was the elimination of double-spends in a *decentralized* fashion; which had been an open problem since the early 1990s.

- Line 360: *Even the smallest change of input (e.g., a single bit) will result in a completely different output digest*

  That is wrong. If the input text is longer than the digest, it is mathematically certain that there will exist different texts of the same length (or longer) that will have the same digest. If the latter has (say) 256 bits, and the input is an English text with a couple thousand words, than the text can be replaced by any other text at all, and the same digest almost certainly could be obtained by inserting a suitable set of extra blanks in the spaces between words.

  What the authors should have said is that, *for the cryptographic hashing functions in use*, there is no *publicly known* and *practically effective* method to generate a text with a given digest, or to modify the input without changing the digest. The only *publicly known* methods to do those things require testing an astronomical number of possibilities, well beyond the capabilities of all the computers in the world.

- Line 383: *The hashing algorithm used (SHA-256) is said to be collision resistant, since to find a collision in SHA-256, one would have to execute the algorithm, on average, about* 2^{128} *times*

  Again, that is not mathematically proved. It is only the effort required to by the best method *publicly known* — namely, just keep computing digests of random texts until a collision is found.

- Line 389: *2.2 Transactions*

  This entire section is superfluous and misleading, since it is very specific to cryptocurrency blockchains — in fact, to the Bitcoin blockchain. As written earlier, the "transactions" of a generic blockchain could contain arbitrary data.

- Line 428: *Public keys are used to derive addresses, allowing for a one-to-many approach for pseudonymity*

  Again, this detail is specific to the Bitcoin system.

- Line 432: *Asymmetric-key cryptography provides the ability to verify that the user transferring value to another user is in possession of the private key capable of signing the value*

  Again, this detail is very specific to bitcoin. A generic blockchain would probably use cryptographic signatures to verify the credentials and/or entitlements of the authors of transactions; however, any further details would be entirely dependent on the application.

  For example, if a blockchain was to be used to store medical records, cryptographic signatures would presumably be used to ensure that the author of a "transaction" is a qualified doctor and is currently in charge of the patient.

- Line 459: *When it is reported in the news that "Bitcoin was stolen from. . . ," it almost certainly means the private keys were found and used to sign a transaction sending the money to a new account, not that the system was compromised*

  That is the standard reaction of bitcoin apologists and peddlers to such events. One would not expect to see such a "salesman's excuse" a NIST publication.

  The continuing occurrence of such incidents — bitcoin loss due to key loss and theft — proves that these risks are indeed a fundamental flaw **of the system** — specifically, of permissionless decentralized blockchains. Since there is no central authority with power to reverse undesired transactions or replace lost private keys, the consequences of those events are much more severe, often irreparable, in decentralized payment systems.

  The same would be true for non-currency permissionless blockchains. Loss of a private key would mean permanent loss of the right to submit certain transactions. Leakage of private keys would let transactions to be submitted by the wrong person, which would be difficult or practically impossible to delete; and there would be no way to prove that the the author was not the original holder of the private key.

  The latter in fact is a known and unfixable flaw of asymmetric signatures. It stems from the fact that there is no sure way to tell whether some information, that was supposed to be "secret", has in fact been copied by some unauthorized person.

- Lines 470,481: *Centralized ledgers* [. . .] *may be lost or destroyed; a user must trust that the owner is properly backing up the system* [. . .] *the blockchain ledger will be copied and distributed among every node within the system*

  As discussed in secton 1.6 of this review, a blockchain does not by itself ensure that all the data

will always be widely replicated and available to everyone who should consult it. Conversely, wide replication can be achieved using traditional databases, too.

- Line 473: *The transactions may not be valid; a user must trust that the owner is validating each received transaction*

  Blockchains do not make a difference in this regard. A centrally managed system can be expose its database and logs to the public, and therefore the validity of its current state and update history can be checked by others just as well as if it was a blockchain. Conversely, a blockchain-based ledger can be made opaque in order to protect confidentiality, and then it may not be verifiable by any observer.

  For example, some cryptocurrencies were designed to hide the amounts of coins moved by each transaction from anyone except the parties involved. A bug was discovered in the software used by some of those coins (including Monero and DashCoin), that would have allowed a malicious user to create infinitely many new coins for himself. As an unintended side effect of the privacy, it would be impossible for general users to notice such a violation of the intended behavior [8].

- Line 475: *The transaction list may not be complete; a user must trust that the owner is including all valid transactions that have been received*

  Blockchains do not make any difference in this regard either. A blockchain only provides (weak) evidence about transactions that have been included in it by the miners. It cannot show whether a valid transaction was received by the miners but was discarded by them.

  The Bitcoin (Core) system is now saturated, and suffers recurrent "traffic jams," with backlogs of hundreds of thousands of unprocessed transactions that take weeks to clear. If the backlog exceeds the capacity of a miner's queue, he will discard lower-priority transactions. On at least two separate occasions, tens of thousands of transactions were discarded by all the miners, for that reason, and never confirmed (unless the affected users noticed the problem and re-issued them).

- Line 477: *The transaction data may have been altered*

  Once more, the integripty of transactions is not specific to blockchains.

  In the bitcoin network, transactions cannot be tampered with, before being confirmed, because the whole transaction is cryptographically signed by the issuer(s). This feature can be (and is) used in traditional centralized ledger systems to prevent tampering of user requests by third parties.

  (As a matter of fact, because of a quirk of the signature algorithm used by the current version of the bitcoin protocol, a bitcoin transaction *can* be modified by a third party while in route to the miners. The interloper cannot change input and output addresses, nor the values transferred; but it can change the transaction's ID. That may confuse some wallet management tools, which may rely on transaction IDs to locate their transactions in the blockchain.)

- Line 477: *a user must trust that the owner is not altering past transactions*

  A blockchain cannot prevent the alteration of transactions that have already been confirmed, As discussed in sections 1.7 and 1.8 of this review, past transactions can be overriden, or canceled

and replaced by different transactions, or could have their interpretation changed by changing the software.

- Line 487: *New transactions are submitted to a node, which will then alert the rest of the network that a new transaction has arrived*

  In most cryptocurrency networks, there is no strong motivation (much less obligation) for nodes of any kind to forward still-unconfirmed transactions to other nodes. That is in fact one of the flaws of the bitcoin protocol. The same problem is likely to exist in any systems based on permissionless blockchains. Even in permissioned blockchains, transactions sent to a miner may be lost if the miner crashes before forwarding them.

- Line 500: *Whenever new users join the system, they receive a full copy of the blockchain*

  As explained before, not all nodes may want or need to receive the whole blockchain. In a permissionless blockchain system, only new *miners* need to do so; and they do not need to store the full blockchain, once they have extracted from it the set of unspent outputs.

- Line 534: *a data structure known as a Merkle tree is utilized*

  The use of a Merkle tree to organize the transactions in a block is a peculiar choice of the Bitcoin protocol, and not an essential feature of a blockchain.

  Indeed, the report fails to explain *why* Satoshi chose that organization for the block's contents. The reason was to let simple ("lightweight") users *efficiently* verify that a transaction had been included in the blockchain, without having to trust a miner or some other server. To do that, the user needs only the chain of headers of the blockchain, the chain of hashes of the Merkle tree that connects his transaction to the block's header, and the hashes that are siblings to those. At present, the user would need to receive only 60-70 Merkle tree hashes, instead of a whole 1 MB block. While the user still depends on other servers to obtain this information, once he receives it he can validate it without having to trust those servers to be honest.

  Therefore, the Merkle tree is only an optimization, that is effective only if the blocks are expected to contain thousands of transactions. If each block is expected to have only a hundred transactions or less, the gains for simple users are probably not worth the added complexity.

- Line 576: *it is difficult to compute a valid block, but computationally easy to verify one*

  This is incorrect. It is not particularly difficult to "compute" (create) a valid block; in fact, it may be even easier than validating one. What is difficult is forging a signature of an address *without the corresponding private key.*

- Line 588: *Lightweight nodes are generally found on smartphones*

  Lightweight nodes are not *found* on smartphones; they *may run* on them.

  However, in a system that serves millions of users, downloading the full blockchain may be so expensive that almost all users will run lightweight software — even if they use powerful desktop computers.

- Line 589: *Lightweight nodes are generally found on* [. . .] *Internet of Things (IoT) devices*

  As far as I know, there is no IoT application that looks amenable to use blockchain technology. The bandwidth, processing, memory, and availability required to run even the simplest lightweight wallet software are probably excessive for the typical IoT embedded computer. (Even the chip produced by the company "21" did not run a lightweight user software [12]. It was only a "dumb" hashing chip, similar to those used in cryptocurrency mining rigs, that depended on the company's centralized mining pool server.)

- Line 592: *Proposed transactions within a blockchain system are stored on mining nodes within an unspent transaction pool*

  That is a major error: it is conflating the *unconfirmed transaction queue* (the *mempool* in bitcoin jargon) with the *unspent output set* (or *UTXO set*). These are two totally distinct databases that serve distinct purposes. (This is not a mere typo, since the error occurs elsewhere in the draft, e. g. on line 805.)

  The report also fails to note that each miner has its own version of the unconfirmed transaction queue, which usually differs from that of any other miner. For example, if a miner goes offline for a few hours, his mempool will probably be missing any transactions that were issued in that meantime. If that miner happens to solve the next block, those transactions will not be in it, and will be left for other miners to confirm.

  This fact is often a source of confusion for bitcoin users. There are sites that ostensibly show "the" mempool, but in fact they merely show their own mempools — which may not match the mempool of any miner. Those sites eavesdrop on transactions issued by clients and then try to emulate the miners' mempool management algorithms.

- Line 607: *Once a puzzle is solved with a particular nonce, the node creates a hash of the block's data and stores it within the block itself*

  That is incorrect. The hash of the block's content is the root of the Merkle tree, and must be computed *before* the miner starts to solve the puzzle. The hash of the *header* is what is informally called "the block's hash", and it does not need to be stored in the block itself.

- Line 617: *many mining nodes are competing at the same time to solve a puzzle to gain the right of publishing the next block (and if applicable, a financial award). They are generally mutually distrusting users that may only know each other by their public addresses. Each user may be motivated by a desire for financial gain, not the well-being of the other mining nodes or even the network as a whole*

  As explained in section 1.9 of this review, a critical assumption of the bitcoin protocol (and of most permissionless blockchain protocols) is that most of the miners are not only motivated by financial gain, but in fact are greedy and only care about maximizing their chances of earning *the next block* before other miners.

  That assumption would be reasonable if mining power was dispersed among thousand of anonymous independent miners, as Satoshi imagined. In that situation, the best strategy for each miner

is indeed the greedy one. However, as discussed in that section, mining of permissionless cryptocurrencies inevitably becomes centralized, so that assumption may easily fail to happen.

- Line 649: *4.1 Proof of Work Consensus Model*

  The authors fail to stress that the proof-of-work model is only applicable to permissionless *cryptocurrency* blockchains — because it requires rewarding anonymous miners, and that can only be done if the system has its own valuable token.

- Line 663: *Hashing a candidate block one thousand or one million times (with different nonce values) only increases the likelihood of solving the current puzzle (as the nonce input space is being reduced with each hash calculation)*

  This is not quite correct. Since the header (80 bytes) is longer than the SHA-256 output (32 bytes), it is believed (but not proved) that the hashes of the same header template with different nonces can be considered *independent* random strings, for the purpose of the Bitcoin proof-of-work. In that case, the the number of nonce values that solve the proof-of-work puzzle is not fixed. In fact, there may be no nonce value that does it. (If a miner exhausts all possible nonce values, it must do some other trivial change to the header, like the timestamp, and start again.) As a result, the chance that a nonce value will solve the puzzle is practically independent of whether a billion other values did or did not solve it.

- Line 695: *There is no shortcut to this process*

  Since the process depends on the SHA256 hashing function, one cannot say that. The best one could say is that there is no *publicly known* shortcut *that would enable the puzzle to be solved at negligible cost*; and that most cryptography experts *hope* that there is none.

  But in fact there *are* shortcuts, some of which have been published and even patented. A famous one is the ASICBoost method, that in theory could reduce the cost of solving the puzzle by 10% or more, by optimizing the computation of the header's hash [5].

- Line 727: *4.2 Proof of Stake Consensus Model*

  The authors fail to observe one little detail: at present, there is no significant cryptocurrency that uses the proof-of-stake (PoS) model. So far it is only an idea, that has been used only by a few cryptocurrencies with no significant e-commerce use.

  Moreover, like proof-of-work, the proof-of-stake method can be used only for permissionless *cryptocurrency* blockchains, because it depends on miners having financial interest (the "stake") in the integrity of the system. If the blockchain is permissioned, the central entity can choose which branch of the blockchain is the official one. If the blockchain is permissionless but has no associated valuable token, the stake of anonymous miners cannot be determined.

  In fact, even for permissionless cryptocurrency blockchains, the PoS method has many unsolved problems — such as encouraging the hoarding of the tokens, thus frustrating their utility as a currency; and promoting the centralization of token ownership, since those who have more tokens will collect more rewards.

- Line 761: *4.3 Round Robin Consensus Model*

  As observed in section 1.2 of this review, permissioned ledger systems require a central authority, and therefore have no reason to use of a blockchain. The round-robin system only increases the role of the central authority, since it must constantly orchestrate the process.

  For example, if the miner of turn takes too long to publish its block, the central authority would have to decide whether he should be excluded from the miner set, and the next miner should be given the task instead. Then, once the laggard miner comes alive again, the same authority must decide whether it should resume its post, and in which round. Without the central authority, some miners may consider the block "too late" and exclude that miner, while others may receive the block in time and thus keep him.

- Line 776: *it is possible that multiple blocks will be published at approximately the same time*

  This section is misplaced, since block collisions are a concern only for permissionless blockchain systems that use proof-of-work. Other conflict resolution methods have other kinds of race problems.

- Line 801: *Most blockchain systems will wait until the next block is generated and use that chain as the "official" blockchain, thus adopting the "longer blockchain"*

  That is not quite correct. First, users and miners cannot "wait until the next block" $n + 1$ when a conflict for block n occurs, because they cannot determine that a conflict occurred until they receive the second of the two blocks with number n. Until then, they must assume that the block n that they received is the tip of the "official" blockchain.

  Indeed, in permissionless cryptocurrency systems like bitcoin, miners are incentivized to accept the first block number n that they receive, and *immediately* start mining the next block $n + 1$ using that one as the parent.

- Line 815: *A soft fork is a change to the technology that will not completely prevent users who do not adopt the change (e.g., an update to the latest version) from using the changed blockchain system*

  This claim is often repeated, and is a tenet of faith of some developers — notably the team who is currently maintaining the most popular version of the software (the "Core" implementation). However, it is wrong.

  In a soft fork, lightweight users who do not upgrade can still receive payments, and will consider valid any branch of the blockchain that is valid by the new rules. However, they may be unable to issue payments, if their wallet software happens to generate transactions that are valid by the old rules but violate the new ones. In the example given by the authors, any user who, for some reason, used the opcode `OP_NOP2` in a script, believing it to be a no-op, would never see his transaction confirmed — and would have no feedback from the system explaining why.

  Moreover, after a soft fork becomes effective, miners who fail to upgrade their software may lose work, if they inadvertently assemble blocks that violate the new rules. This happened, for example, in the "Fork of July" incident in July 2015, when a small miner failed to upgrade in time for the BIP66 fork, and solved a block that was invalid by the new rules.

By the way: even though the authors take care to distinguish between *miners*, *full* (but non-mining) *nodes*, and *lightweight nodes*, they are not careful when using the nomenclature in the rest of the draft. They often use "node" when they should say "miner", e.g. on line 818.

- Line 817: *a soft fork can be backwards compatible, only requiring that a majority of nodes upgrade to enforce the new soft fork rules*

  That is not correct. A soft fork will succeed if, and only if, a majority of the **miners** (counted by mining power) starts to use the new rules. If that happens, all the lightweight users, all full but non-mining nodes, and all the remaining miners will be forced to upgrade, as explained above.

  On the other hand, if only a minority of the miners accept the new rules, then the blockchain will almost certainly and permanently split into two independent branches, each functioning as a separate ledger. The longest-chain rule will **not** resolve the conflict. All the nodes that failed to upgrade — including the remaining (majority) miners — will eventually accept the "old" branch, because it will eventually become the longer one; while those nodes that upgraded to the new rules will ignore that branch, as being invalid, and accept the "new" one.

  To avoid the risk of splitting the chain, the Core developers have introduced a mechanism in the protocol that lets miners signal, through bits in the header of their solved blocks, whether they intend to accept a proposed soft fork. The Core developers then arrange in the software that the rule changes will be activated only after a substantial majority of the recent blocks signal acceptance.

  However, nothing in the protocol ensures that the miners who signaled acceptance will continue mining after the fork, and they may change their mind and fail to actually accept the new rules. Thus, the voting mechanism is a solution to the "risk of chain split" problem only the hacker's sense of the term — not as responsible software engineers and computer scientists understand it. Indeed, the mechanism failed in July 2015, causing the so-called "Fork of July" incident (a rewind-and-rewrite event spanning six blocks).

- Line 827: *A hard fork is a change to the technology that will completely prevent users who do not adopt it from using the changed blockchain system*

  The authors should have written "validity rules" instead of "technology".

  Anyway, that claim is incorrect. Depending on the nature of the changes, users who did not upgrade may still be able to issue transactions to users who have upgraded, and vice versa. However, fully-verifying nodes that did not upgrade may ignore blocks solved by upgraded miners (and any blocks that descend from them). Lightweight nodes may or may not ignore such blocks, depending on the extent to which they verify them.

- Line 830: *Users on different hard forks cannot interact with one another*

  First, the writing is confusing: the authors should have written "users who adopt the hard fork changes cannot interact with those who don't, and vice versa".

  However, the claim itself is incorrect. Depending on the change in the rules, some transactions may continue to be valid under both versions of the rules. Therefore, if the chain splits, such transactions can be confirmed in both branches of the chain.

It can be argued that this situation is undesirable, because it would get users confused; therefore, many bitcoin experts recommend that hard-fork changes are designed so as to make such "replays" impossible – that is, so that every transaction that is valid under either version of the rules is always invalid for the other version, and cannot be modified to fit the latter except by the user(s) who issued it.

Moreover, if a chain splits, a user (person) can always run both versions of the wallet software, with two separate wallet files; and thus send payments to other users of either version. What it cannot do is transfer coins from one branch to the other (except by trading them in some exchange service).

The authors seem to be unaware that whether the blockchain splits or not does **not** depend on whether the fork is "soft" or "hard", or on the choices of non-mining nodes, but on the nature of the changes and on how many miners accept them. As explained above, a soft-fork type of change (that only hardens the validity rules) will split the chain if, and only if, the miners who accept it are a minority. A hard-fork change that only relaxes the validity rules will **not** split the chain if, and only if, the miners who reject it are a minority. In any case, the blockchain chain will not split (for practical purposes) if virtually all miners accept the change, or virtually all of them reject it.

- Line 873: *6 Smart Contracts*

The authors fail to mention all the major flaws of the "smart contract" concept, that render it useless in practice.

For one thing, a smart contract system implemented by a blockchain ledger cannot use any information that is not recorded in the blockchain itself. That restriction is necessary because, by definition, any third party must be able to verify that the contract has been correctly executed, at any later time; and reach the same conclusion as any other user who does the same. If the contract used some external source of information, then a change in that source could invalidate previously valid blocks.

The only external information that a smart contract platform can use is information whose validity can be ascertained without reference to external sources — such as transactions submitted by users, or allowed choices made by miners when assembling their blocks. Thus, a smart contract may specify "if at least 10 coins are sent to address X, then activate smart contract Y;" but it cannot say "if the car is returned in good state, refund the rental deposit."

Another major flaw of smart contract systems is that they are not legally binding. Thus, a smart contract cannot say "if 10 coins are sent to address X, send a plasma TV to this person." If the 10 coins are sent, the system will be unable to force the merchant to deliver the TV.

The guy in that example would have to resort to courts to have the contract enforced — thus negating the main alleged advantage of smart contacts. However, he would have to sue in the jurisdiction of the merchant, and he would have to convince the court that the smart contract was approved by the merchant, and not submitted to the system by someone else without his approval.

In any case, the courts are unlikely to accept being subordinated to a computer system that is supposed to be immune to any legal system.

Because of these limitations, smart contracts have yet to find effective practical uses. Their main

use so far has been to create financial schemes, like the DAO and the unending stream of ICOs, that aim to trap money from gullible investors into complicated mechanical mazes that, in the end, will deliver much of that money to the scheme's creators.

- Line 877: *The* [smart contract] *code, being on the blockchain, is immutable*

  As discussed in section 1.8 of this review, blockchains are not immutable, and modifying past blocks need not be more difficult than modifying past entries in a centralized ledger implemented with established technology. (This claim is inexcusable, considering that the authors cite the reversal of the DAO hack, on line 833, as an example of a fork.)

- Line 881: *For example, the authors of this document have created smart contracts that publicly generate trustworthy random numbers*

  I did not have the time to read the description of that smart contract. However, I hazard the guess that a miner with a significant fraction of the total hashpower could "load the dice" of that generator by refraining to publish a block that he solved, if doing so would cause the generator to issue an unfavorable outcome

  For example, suppose that a miner with 10% of the total hashpower bets $50,000, on even money, that the last bit of that random number will be '1'. That number presumably depends on several mined blocks. When the last of those blocks is to be solved, the miner has 10% chance of solving it. If he sees that, with his solved block, the bit turns out to be '0', he just discards the block. Then, even if he fails to solve the block again, that bit will be '1' with probability $0.90 \times 0.50 + 0.10 \times 0.50 + 0.10 \times 0.50 \times 0.50 = 0.525$.

- Line 939: *7.1.2 Use Case Examples* [of permissioned blockchains]

  It is revealing that none of the "examples" described in this section is a real permissioned blockchain system that has been actually built and deployed. They are just "exciting ideas," dreamed up in five minutes by blockchains enthusiasts — who have never been able to flesh out the details.

- Line 954: *Recording* [in a permissioned blockchain] *the transfer of physical goods from a producer, to a shipping terminal, to a ship, to a cargo train, to a delivery truck and to a store is an appealing application of blockchain technology. A blockchain could play a crucial role in trust and transparency with end customers*

  Using a blockchain to record that information, instead of any other database structure, would have absolutely no influence its trustworthiness or transparency. One can record in a blockchain the shipping of 500 fresh unicorn eggs to LA, while actually sending a bucket of pebbles to NY. To prevent fraud in such an application, one needs to ensure that the data entered matches the reality. That goal can only be achieved through "physical" measures — like restricting data entry to properly trained and trustworthy persons, holding frequent and thorough inspections, ensuring prompt and effective punishment for fraudsters, etc.. None of these measures depends on the technology used to record the data.

- Line 984: *Since anyone could contribute to the* [permissionless] *blockchain, some could submit false data to the blockchain, mimicking data from valid sources. Is there a way for the application to ensure it only gathers data from reputable sources?*

  The authors could have helped the readers by providing the obvious answer for this question: "No."

- Line 987: *Many applications follow the "CRUD" (create, read, update, delete) functions for data. With a blockchain, there is only "CR" (create, read).*

  As discussed in section 1.5 of this review, the authors (and most proponents of the technology) fail to realize that blockchains are not alternatives to the "CRUD" live databases of centralized systems, but only to the "CR" historical logs that those systems should keep anyway for auditing, backup, or statistical purposes.

- Line 1021: *8.1 Cryptocurrencies*

  This section is totally superfluous and inadequate. For one thing, it fails to inform the reader about the historical, political, and financial factors that dominate the evolution of each coin. For example, there is no mention of Bitcoin's so-called "block size war," the bitter dispute for control of the "Core" software repository in GitHub, the splitting of the community and of its forums, the disputes between the Blockstream developers and the Chinese miners, the actions of Litecoin's creator about is creature, etc. Anyone contemplating the use of a cryptocurrency blockchain for some other application must absolutely be aware of these factors.

- Line 1030: *8.1.1 Bitcoin (BTC)*

  The developers of the "Core" implementation naturally insist that the cryptocurrency defined by their software is the only legitimate "Bitcoin". That claim is echoed by other people who are bound to that cryptocurrency in some way — such as investors and managers of "Bitcoin funds" which can only hold one currency and have opted for BTC.

  However, the fact is that the single "Bitcoin" cryptocurrency that existed until 2017-08-01 has split into two active currencies. While both are active and largely similar to the old Bitcoin, neither is identical to it: on that date, both branches introduced changes in the validity rules, that had significant impact on the users.

  Therefore, in the interest of clarity and accuracy, one should always use a qualified name when referring specifically to one of the two branches. For the branch that uses the "Core" implementation, the qualified name most used outside its supporter community is "Bitcoin (Core)" or "Bitcoin Core." A NIST report should use that qualified name, ignoring the objections of its fans.

- Line 1055: *8.1.2 Bitcoin Cash (BCC)*

  The three-letter symbol generally used for Bitcoin Cash, including by its developers and supporters, is now "BCH". The symbol "BCC" was briefly used when the currency was created, but had to be abandoned because it was already in use by the (now defunct) BitConnect pyramid scheme.

- Line 1123: *8.2 Hyperledger*
- Line 1155: *8.3 MultiChain*

  These two sections read like a list of commercial advertisements for products, with texts that could have been drawn from the respective press releases.

- Line 1175: *there is still a group of core developers who are responsible for the system's development*

  There is no such thing in the permissionless blockchain concept.

  Each protocol may have many implementations. Being open projects, most cryptocurrency implementations accept code conttributions from anyone, and can be cloned and modified by anyone at will. Some developers may work on more than one implementation. However, each implementation usually has a "head developer" who holds the keys or passwords that allow modification of that code. The head developer decides which patches are included in the official releases of the implementation, and thus indirectly decides who else is in its development team. For example, Wladimir Van der Laan is currently the head developer of the most popular BTC implementation, called "Bitcoin Core" or just "Core". The Bitcoin Cash currency (BCH) has three active independent implementations ("ABC", "XT", and "Unlimited"), each with its own head developer and development team.

  Unfortunately, there is no mechanism in a permissionless blockchain protocol to define which implementation (and therefore which team) is the "official" one. Each miner can run whatever implementation of the protocol he chooses (including his own proprietary one, if he cares to). All miners had better use the same block validity rules, otherwise the chain may split; however, implementations may differ on other observable aspects, such as transaction fee and priority policies, block transmission protocols, handling of potential double-spends, and so forth.

  In fact, there is no reliable way for an external observer to tell which implementation was used to mine a given solved block of a permissionless blockchain. (While the software version stamp may be included in each block, there is no way to verify if that stamp is accurate.)

  Consequently, permissionless blockchain systems have no mechanism to decide whether, when, and how proposed changes to the protocol should be implemented. Several cryptocurrency projects have seen bitter disputes about such changes, even within the same development team. See for example Bitcoin's "block size war" and the imposition of the SegWit change. The three head developers of BCH have struggled to reach agreement on a necessary change to the difficulty adjustment algorithm. (There used to be a fourth implementation, but its head developer gave up after his proposals were not accepted by the other three.)

  Indeed, the lack of an effective *governance mechanism* for protocol evolution is another fatal flaw of permissionless blockchain systems in general.

- Line 1176: *These developers may act in the interest of the community at large, but they still maintain some level of control.*

  Even when a permissionless blockchain system has a single development team who is de facto in charge of its evolution, nothing ensures that the team's decisions will be guided by the interests of the user community. For instance, the development of most cryptocurrencies is largely determined

by the financial interests of the developers, which in turn depend more on the currency market price than its suitability for payments.

Some teams may steer the evolution of the protocol in a direction that favors certain companies, e.g. holders of certain patents or contracts. This apparently is the case for Bitcoin (Core). Since 2014–2015, the Core development team has been dominated by employees and contractors of Blockstream and associated companies, and the evolution of the BTC protocol has followed a roadmap traced by that company.

- Line 1275: *Blockchains are a significant new avenue for technological advancements, enabling secure transactions without the need for a central authority.* [...] *new applications beyond the realm of currencies are building upon the fundamentals of blockchain technology.*

  These claims are incorrect. Blockchains are not really "new", but rather "primitive". They are actually less secure than established replicated database technology, as generally used by credit cards and banks. Permissioned blockchains still require a central authority of some sort. Permissionless blockchains must be cryptocurrencies in order to reward the miners, and then mining inevitably becomes centralized too.

- Line 1307: *Blockchain technologies have the power to disrupt many industries. To avoid missed opportunities and undesirable surprises, organizations should start investigating whether or not a blockchain can help them*

  This concluding remark fittingly summarizes the main flaw of the draft report. The alleged "disruptive power" of blockchain technologies is only a fantasy of the self-declared"blockchain specialists," that is not supported by a single example of successful application in spite of several years of intensive search for one. Until such an example arises, organizations should not waste time considering the use of this technology.

3 Conclusions

When deciding the fate of that draft report, the NIST editors should also consider of the context in which it was produced. There may be hundreds of projects — in the form of startups, or working groups within established companies — that propose to apply blockchain technology to all sorts of things, from dating [7] to the tracking nuclear weapons [2]. Most (if not all) of those projects are staffed by enthusiastic but unqualified young people, often fresh out of college or school drop-outs, who have no experience in databases, distributed services, networking, or financial computing.

Those "instant experts" don't see the need to acquire such knowledge, because they assume as an axiom that blockchain technology is revolutionary and has rendered obsolete all existing technology in those fields. For the same reason, they do not see the need to understand the real-world application that they intend to address, since they are sure that blockchain technology will radically change its very nature. Thus, they are confident that the knowledge that they acquired about blockchains, no matter how superficial, is all that they need in order to succeed.

That mindset unfortunately permeates the whole draft; and official endorsement of the report by NIST would only bolster it.

As noted above, blockchain technology has yet to prove its worth in *any* application, including the one for which it was originally developed. Thus, a NIST report on the principles and uses of blockchains would be as well-founded and necessary as that famous report by the US military on how they would handle a zombie epidemic. Except that this report would not be as fun to read as that one.

In view of the points above, I strongly advise against the publication of the draft as a NIST report. It would be a disservice readers interested in blockchain technology, because of its many errors and omissions. It would be a disservice to those who need to implement some sort of ledger, because its biased outlook will lead them to waste time considering a technology that, almost certainly, will be inadequate for their needs.

It the NIST feels that it really owes to the public a report on the technology, I suggest that its contents be reduced to three words: "Blockchains are useless."

References

[1] Andreas Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. 2nd edition, 2017.

[2] Aaron Arnold. Blockchain: A new aid to nuclear export controls? thebulletin.org/blockchain-new..., 2017.

[3] Bitcoin wiki: Protocol documentation. en.bitcoin.it/wiki, 2018.

[4] David Gerard. *Attack of the 50 Foot Blockchain*. 2017.

[5] Timo Hanke. Asicboost - A speedup for Bitcoin mining. ArxXiv paper 1604.00575, arxiv.org/pdf/1604.00575.pdf, 2016.

[6] JoelKatz. Unfortunately, due to a server bug, some history was lost. .... bitcointalk.org topic 174854 msg 2352658, 2013.

[7] Luna: Blockchain-optimized dating. www.meetluna.com, 2018.

[8] Disclosure of a major bug in CryptoNote based currencies. getmonero.org/2017/05/17/disclosure..., 2017.

[9] Strange block 74638. bitcointalk.org topic 822.0, 2010.

[10] 11/12 march 2013 chain fork information. bitcoin.org/en/alert/2013-03-11, 2013.

[11] A complete history of bitcoin's consensus forks. blog.bitmex.com, 2017.

[12] Jorge Stolfi. How to get $100 million in vc funding to build an industry that makes $300 million profit without spending a dime. www.reddit.com/r/Buttcoin, 2016.