SoK: Bitcoin

(Systematization of Knowledge Track)

Authors Removed for Blind Review

***Abstract*—We provide the first systematic exploration of the design space of cryptocurrencies, including Bitcoin, the "altcoins" that compete with it, and enhancements proposed by the open source development community and academic researchers.**

I. Introduction

Bitcoin is a digital currency that has recently become very popular. [AM: more backstory?] It has received some attention from researchers, who have analyzed aspects of its operation and proposed extensions and solutions to attacks. A large and vibrant community of open-source developers and other participants have stewarded the system, proposing (and implementing) numerous extensions and modifications as well. Some of these have been implemented as alternate protocols, and compete with Bitcoin in a larger ecosystem of so-called "cryptocurrencies", while others have been adopted into Bitcoin itself (which remains, as of this date, far and away the most popular). The primary goal of this paper is to provide a framework for understanding and evaluating points in the space of possible designs for Bitcoin and related systems.

Bitcoin's goals are nominally focused on payments and the maintenance of a virtual currency - a user who owns a quantity of the currency may transfer it to someone else, typically in exchange for goods or services. This goal itself is not especially novel or difficult, especially if it's acceptable to rely on trustworthy servers and administrators. Digital currencies like Linden Dollars and World-of-Warcraft gold already provide this functionality; so do payment systems such as Paypal or debit cards (denominated in ordinary state-administrated currency, like the US Dollar or the Euro). What's remarkable about Bitcoin is that it aims to do this in a difficult setting *without any trusted parties* and *without pre-assumed identities among the participants*. Instead, the premise of Bitcoin is that greed is predictable[AM: incentives can align?] - given the right incentive mechanism, the network can encourage enough participation from the anonymous public to withstand any malicious but resource-constrained attacker. [AM: Basically it's the difficult model that makes Bitcoin interesting, not the money itself]

The contributions of Bitcoin are quite surprising here. The general problem of consensus in a distributed system is impossible in an anonymous network. Cryptographic approaches to secure multi-party computing rely on pre-established identities or a trusted PKI. Incentive schemes for p2p networks have typically assumed an external bank exists, rather than the ambitious approach of implementing money from scratch internally. Even approaches to cryptographic cash (ecash, micromint) rely at least on a trusted bank or issuer. Bitcoin has survived its first five years without suffering any critical attack - despite the clear temptation of breaking it in a way that steals the money - although it has not yet been given a formal theoretical foundation. The original whitepaper introducing Bitcoin sketched a proof that it is secure in an "honest majority" distributed system model, although this doesn't illuminate under what conditions, if any, the built-in incentive scheme *leads* to a sufficient amount of correct participation. In this paper, we do not try to provide a theoretical model of Bitcoin. However, in providing a more abstract understanding of Bitcoin, we believe our contribution can help inspire a theoretical development. [AM: The gap in theoretical basis makes it interesting, and is part of our motivation.]

If Bitcoin is successful in agreeing on a transaction history and replicated state machine, then it solves a much more general problem than merely transmitting payments. Indeed, there have been many proposals have been for more complex financial transactions than simply transmitting money, and there have even been proposals for non-financial transactions. One key to our understanding of the design space of cryptocurrency is the notion of "Bitcoin as a platform" for transactions involving arbitrary rules. The abstract description we arrive at, surprisingly, does not include a particular definition of money as a goal - instead, money shows up as part of a particular solution.

A. Related Work

- Other digital currencies, centralized like QCoin, Linden dollars. Other payment systems, like Paypal, debit card systems, etc.

- Quick summary of major academic work on Bitcoin, including extensions of Bitcoin, such as zerocoin. Other surveys or introductions to bitcoin, such as the bitcoin primary, etc. Analysis of Bitcoin, such as Kroll, et al.

- Other systems for distributed consensus, Byzantine impostors, homonyms, FLP. Preliminary approaches to ecash, chaum, micromint, hashcash. Proof-of-work, client puzzles.

- Cypherpunks, Wei Dai, Nick Szabo, etc.

II. An Introduction to Bitcoin

[AM: There are a few conflicting goals in this section. One is to introduce the basic terminology and concrete details about how the present Bitcoin works. Another is to introduce the key abstractions we'll use to understand Bitcoin variants. Another is to understand Bitcoin in context.]

A. An Alternate History of Bitcoin in Steps

[AM: The purpose of this section is to explain what's new about Bitcoin, and how it might have been derived systematically as an improvement to other approaches.]

The goal is to have basic checking account functionality for Alice and Bob.

a) Step 1: Imagine theres a trusted third party (TTP): Each participant is (somehow) assigned a single key pair which also acts as an address. The TTP allows anyone to send it messages and gives all participants a consistent view. Each participant (i.e., address) has some balance. We will defer the question of initial balances and where these balances come from.

If Alice (address A) wants to send Bob (address B) value v, she sends to the TTP a signed statement, called a transaction, saying address A transfers value v to address B. The TTP verifies each transaction (checks that $vbalanceA$) and ignores the ones that dont verify.

b) Step 2: Removing the TTP: If Alice (address A) wants to send Bob (address B) value v, she broadcasts a signed statement to all other participants saying address A transfers value v to address B. Once every 10 minutes, the participants themselves (replacing the TTP) conduct a protocol to a) propose a batch of valid transactions, called a block, and b) vote on these proposals to select one.

Well known voting protocols have been studied, under some standard assumptions. First, it is assumed that a majority of the participants are honest. Second, its assumed that honest parties are able to broadcast synchronously to every other participant. Dishonest participants, on the other hand, are assumed to be able to equivocate, sending one message to some participants and a different message (or nothing) to others. Underlying all this is the fundamental assumption that there is a fixed set of participants that is known to everyone.

[1]

c) Step 3: Removing limits on addresses: In reality, participants dont have preassigned identities, so the assumption that there is a fixed set of participants is untenable. Therefore we replace that assumption with a much weaker one: Imagine that theres a magic token that lands on a random participant (chosen uniformly) at regular intervals. [2]

As before, participants broadcast each transaction to all other participants. When a participant gets the token she collects all the transactions shes seen (in the most recent token interval), verifies them, and broadcasts the block.

[AM: Description of the hash chain data structure, how participants process blocks, longest chain rule, and how this leads to probabilistic consensus if the majority of the identities are honest.]

Note that participants can now create key pairs for themselves – as many as they like. We have severed the link between participants and addresses. The only assumption is that the token selects participants uniformly at random, and hence selects an honest participant with a probability greater than $\frac{1}{2}$.

d) Step 4: Removing the token: proof-of-work: Participants compete for possession of the token by racing to solve a proof-of-work based on hashing. Thus, participants acquire the token with a probability proportional to the fraction of Bitcoins hash power that they control. [3] Attaching the proof-of-work to the block proves possession of the token, which is purely imaginary.

Now, instead of the majority of participants being honest, we require that the majority of hash power be controlled by honest participants. Except for that change, the argument for consensus in Step 3 holds here as well.

e) Step 5: Incentives: To better justify the assumption that a majority of participants (by CPU power) are honest, we provide an incentive for honest behavior. This is done by adding value to the balance of an address chosen by each participant that finds a proof-of-work solution. [4] [5]

This suggests a bootstrapping argument. If the transaction log is secure (has integrity and availability), then it will be useful as money, and therefore value added to an account makes a worthwhile incentive. This in turn will encourage a large amount of participation, which ensures the transaction log is secure.

This appears to be a virtuous cycle. On the other hand, neither system can exist without the other. If the integrity or the availability of the transaction log can be easily attacked, then we cannot expect it to maintain much value as a currency. Likewise, if the currency is not valuable as money, then it will not be an effective incentive for encouraging sufficient participation to prevent successful attacks.

B. Law, Log, and Index: An Abstraction of Bitcoin as a Platform

Abstractly speaking, what functionality does Bitcoin – or more importantly, the innovations embodied by it – provide? Originally presented as a mechanism for online "payments," in the abstract it is a much more versatile platform.

1) (Log) A sequential broadcast medium. Every user eventually (in a short amount of time, with high probability) agrees on a global sequential log of messages (and every user receives every message in its entirety). Any user can publish a message, and it will be included in the log within a short time (although, a monetary fee may be required for timely service).
2) (Index) Participants on the network maintain a globally replicated storage index. Every message in the broadcast log is processed, and in some cases causes updates to the index. These indexes can be relied on as concise summaries of the public broadcast record.
3) (Law) The broadcast log may additionally contain data that reflecting the result of computations, including queries to the index.

This framework can be used to understand the design differences between many Bitcoin-related proposals, all of

[1] For efficiency, there are randomized protocols.[AM: survey of randomized consensus, breaking n^2 barrier]

[2] The magic token is recognizable and unforgeable: if a participant receives a message from a participant holding the token, she knows that the sender did in fact hold the token.

[3] there is a random nonce, so that each participant works on a different part of the problem space.

[4] This is also the mechanism by which the values money are initially distributed, in an arguably fair way.

[5] Everything described so far is merely an abstract system for transferring and tracking balances. For the incentive system to work, these balances must be interpreted as money.

which are special cases of the general structure. As an example, the standard usage of Bitcoin can be described as the following instance:

1) The broadcast log is used to publish signed messages called "transactions", which reflect an intent to transfer quantities of currency from one account to another. Quantities of the currency may also be set aside as fees.
2) Participants maintain a *ledger* containing a mapping between every quantity of currency and the public key of its current owner. This index must be updated to reflect the new balance after every transaction.
3) Transactions are considered "valid" only if the attached signatures are valid with respect to the public keys in the current ledger, and if they spend a reasonable amount of money. In fact, only transactions that are valid according to these rules may be included in the log (although arbitrary data — including invalid transactions — may still be published as auxiliary data contained within a valid transaction).

All three of these components are necessary, and complement each other. The sequential log is used to ensure that no two transactions double-spend the same quantity of currency. On the other hand, since only valid transactions carry fees, the index is used to ensure that data needed to validate transactions is readily available.

Since only valid transactions are included in the log, end-users (such as merchants with mobile devices) need only check that a transaction is included in order to confirm a payment. Although the log necessarily grows without bound, the mutable index does not; therefore nodes can "prune" old data from their main disks. These optimizations are discussed in more detail later on.

Most proposals for Bitcoin-like variations, including overlay currencies such as Mastercoin and Colored Coins as well as more ambitious altcoins like Ethereum, can be described in a similar way as we will show in Section anoterefer. Additionally, these key components provide a basis for evaluating the resource complexity of various schemes and use-cases.

The view of Bitcoin as a platform is that the currency is only inherently useful to the extent that it allows you to pay for transaction fees. In this sense it is comparable to postage stamps.

C. System Attack Model and Consensus Requirements

The Bitcoin mining consensus proof from the whitepaper, assuming a majority of the hash power is "honest". Bitcoin's solution to the consensus problem is novel, although the proof known so far is for the wrong model (honest rather than altruistic). Given a set of rules, in a distributed system, the problem remains of how to choose a correct order of operations.

Although, so far, the economic incentive system seems to work, there are only heuristics and no clear model. As best we can tell, the reasoning is circular. Incentive system is needed to make the system secure. The built-in currency that rewards participants has to be valuable for the incentive to be meaningful. The currency is valuable only if the system is secure.[6] Attacks are discussed further in Section[AM: refer].

Bitcoin's assumptions are weaker than in a standard distributed system - most notably, there are no pre-established identities. On the other hand, the assumptions are about the rational preferences of population - specifically that they respond to incentives, even when the incentives are inconclusive. Seems to be circular?

Stabilizing consistency is weaker than a typical distributed system. In a typical distributed system, you receive an acknowledgment at some finite time that the transaction has been committed. In Bitcoin, there is no such acknowledgment, and users must user their own discretion about how long to wait to consider a transaction committed. [AM: This is partially a historical note, meant to clarify how Bitcoin's version of consensus/broadcast relates to more standard notions]

A typical definition of fairness or liveness would specify that *any* transaction should eventually accepted, without reference to adequate payment. However, since Bitcoin operates in a model with no established identities, this would be impossible due to the ability for an attacker to create sybil identities and flood the system.

D. Concrete Details

- Bitcoin's underlying mechanism.

- Mining, transaction propagation.

- Introduction of currency. Use of exchanges, market places. Financial ecosystem.

E. How Changes are Applied

- Discussion of governance and the need for social out-of-band consensus to agree on rule changes. Soft forks, hard forks, and policy. Very few hard forks in history.

- Discussion on the forums and mailing list that provide.

- Ability for altcoins to develop.

F. Towards an Economic Model

- Pooled mining and infrastructure investment.

- A history of pooled mining, and what we can infer empirically about the motivations of miners? Prefer low variance.

III. Breaking Bitcoin

A. Attacks on Privacy

Much of the original focus on digital cash (starting with Chaum [2]) centered on providing anonymity (or pseudonymity) for users. Although Bitcoin was widely reported to provide financial privacy, the system does not in fact include any of the privacy-enhancing technique from that era; instead all transactions are published in plaintext. Thus a large body of work quickly established that Bitcoin transactions are highly linkable. [3][AM: far more citations to add here].

[6]For example, when there have been even temporary lapses in service or bugs announced, the price of Bitcoin has fallen [1].

Although these attacks on user privacy apply to Bitcoin in its current use, there have been proposals to add transaction privacy back in through the use of third-party mixes [4] or integration with more sophisticated cryptographic techniques; these are discussed later.

B. Attacks on Stability

Although the original Bitcoin system came with a proof of security assuming half of the network (by hashpower) follows the protocol correctly, it may be unreasonable to take this as a model assumption. Instead, the choice of users to participate correctly (or at all) are influenced by incentive mechanisms throughout the system.

- The goldfinger attack, and death spiral [5].

- Majority is not enough [6].

- Comment about the threat of altcoins to divide participation and investment, and the CoiledCoin event as an example of this attacking altcoins. [7]

IV. Exploration of the Cryptocurrency design space

Since Bitcoin's inception, a vibrant community of participants and open source developers, as well as a number of researchers, have suggested improvements and extensions. We have developed a framework to aggregate and systematically evaluate these contributions.

Two main categorizations stand out: a) there are proposals for protocols that use Bitcoin (or a variant) as a platform, and b) there are proposals for modifications to the platform itself. We treat these separately.

A. Source Collection: Technical Discussions from the Bitcoin Community

Technical discussions and formal presentation of Bitcoin extension ideas are primarily conducted through the following venues:

- Bitcointalk forums (https://bitcointalk.org/)
- IRC Channels (irc://freenode.net/#bitcoin-dev)
- Developer mailing list hosted on sourceforge
- Bitcoin wiki
- Bitcoin Improvement Proposals

Our methodology involves collecting ideas from these and analyzing them.

As a resource to researchers, the discussions and ideas discussed in these venues represent a) what is desired about Bitcoin from its participants, b) what concerns or as of yet unexplained, c) what is thought to be possible or difficult, tradeoffs, which provide starting points for other research, and d) what facilitates actual deployment.

[7] CoiledCoin was an altcoin that was destroyed by a significant history revision attack from Eligius, a Bitcoin mining pool. https://bitcointalk.org/index.php?topic=56675.msg678006#msg678006

B. Categorization of Extensions

1) Design Tradeoffs: The need to compete with other similar cryptocurrencies establishes the essential tension underlying design decisions and tradeoffs. Bitcoin must encourage as much mining participation as possible in order to defend against powerful attackers. The main mechanism for encouraging miner participation is disbursing rewards denominated in Bitcoins, the internal unit of currency. Thus it is essential for that the application service provided by Bitcoin is useful enough that users are willing to pay usage fees and purchase currency from the miner. If an alternate cryptocurrency offers more features and appeals to more users, then it may compete with Bitcoin for miner participation, thus weakening its security. Because of this, Bitcoin may be expected to incorporating new features and functionalities if they appear to have sufficient user demand.

On the other hand, new features and functionality typically involve increased cost (in either, communication, storage, or computation) to participants running "fully-validating" nodes, which is a requirement for correct mining. It is unclear how much additional burden can be shouldered by the network without a loss of security.

A straightforward approach to reducing the costs of additional functionality is to delegate control to a trusted third party. This is axiomatically discouraged in the design goals of Bitcoin, as it constitutes an "existential threat" that increases the risk of an overall system failure, despite immediate benefits to efficiency and convenience.

Suggested improvements to Bitcoin generally fall into the following categories:

- Efficiency improvements
- Increased stability and security (against various threats to the overall network, or economic stability)
- Additional functionality, to enable new uses.

We prefer to distinguish between two kinds of security: as it concerns the health/stability of network system overall (e.g., as concerns Proof-of-Stake, Block times) and security of users (e.g., ZeroCoin, better keys or password protection).

It's worth identifying that there are many roles in Bitcoin.

- Participation as a node, i.e. relaying transactions and blocks
- Mining, including participation in a mining pool
- Exchanging, mixing, mirroring, merchants, and other services interacting with the currency itself.

A desirable feature may impact some roles but not others. For example, transaction privacy (e.g., Zerocoin) increases the security users but has no direct impact on miners. Likewise, additional costs to miners (or validating nodes) do not necessarily impact users directly.

Due to the need for governance, everyone to agree on the constitutional rules, there are several routes to deploying improvements to Bitcoin. Most proposed extensions to Bitcoin are essentially independent of the mechanism by which they

could be integrated. For example, although Zerocoin is proposed as an extension to Bitcoin, it could also be implemented as a concurrent and independent protocol (an "altcoin"), or as a third party service that Bitcoin users interact with. Other arrangements are possible as well, such as implementing a proposal as an overlay on top of Bitcoin. Another approach, considered a "gradual soft fork" involves having the protocol be voluntarily included by miners. If the network of miners appear to reach a de facto consensus on a new extension, it would be easier politically to incorporate this in the mandatory rules. The following are ways in which an extension may be deployed:

- Hard-forking "mandatory" change, blocks without the modification will be rejected
- Soft-forking changes, for subsets of existing possibility, miners can include cooperating data backward-compatibly with blocks.
- Peer-to-peer network changes.
- Optional client changes only, compatible with existing p2p network.
- An external "oracle," implemented as a trusted third party (or shared/quorum of third parties) recognizable by their public key signatures
- An altcoin, a separate concurrent protocol

This imposes a hierarchy of extensions. In our terminology, an extension to Bitcoin may require only modifications to the client - anything that modifies how a user interacts with Bitcoin can count as a modification. We do mean to rule out things like ATMs that interact with the ordinary Bitcoin client. Bitcoin, even its current form, supports a number of extensible uses. Enables protocols between individuals, use Bitcoin in some way.

On one hand, the reference client accounts for the largest number of Bitcoin nodes. However as many users turn to hosted bitcoin wallets, it is hard to assign an actual user number to these.

V. Protocols using Bitcoin as a Platform

The standard usage of Bitcoin is for financial transactions, in particular payments, denominated in the built-in currency. However there are a variety of more complicated financial transactions, including the construction of other currencies overlayed, as well as other uses, such as timestamping.

Protocols using Bitcoin as a platform can be evaluated in several ways:

- *Validation Costs.* The computational resource burden imposed on the network. This primarily consists of a) the total amount of communications that must be broadcast to the network (i.e., the size and number of transactions involved), b) the amount of storage that the transaction places in the replicated index, and c) the cost of validating a transaction. We analyze these according to the asymptotic cost, per transaction. In addition to the per-contract cost, there may also be an overall cost independent of the per-transaction cost; we indicate this cost in brackets.
- *Party Costs.* Parties that participate in an outer protocol using Bitcoin as a component may have to provide their own additional communication resources as well. This includes local storage, computation, and communication (with either peers in the Bitcoin network or the other parties of the outer party), as well as the amount of time required before the transaction is complete. As with validation, there may be a baseline cost independent of the per-transaction cost; this is indicated in brackets.
- *Trust Model.* While the Bitcoin network itself does not rely on trust in any designated party, outer protocols using Bitcoin are often characterized by leveraging or relying on trust in one of the parties. We indicate these by codes which will be described in the sections
- *Fork Required?* What modifications to the existing Bitcoin protocol (if any) are required.

A summary of this comparison, applied to the proposals we have collected, is provided in Table V-A6.

A. Descriptions of the Rows

1) Purchases: Purchases involve the transfer or exchange of Bitcoin as money, wherein a Buyer orders a good from a Merchant, pays, and then the Merchant delivers the good. **Merc** indicates that the Merchant must be trusted to complete the delivery; **3rd** indicates that a third party, separate from the Buyer and Merchant, is trusted to intermediate.

- (Purchase, SPV) In a standard Bitcoin transaction, a Buyer transfers a quantity of BTC to a Seller by publishing a transaction referring to a coin currently owned by the Buyer, and rededicating it to a public key owned by the Seller. The Seller waits for some number of confirmations, until he is confident that the block containing the transaction will not be revised, and then performs an irrevocable action, such as shipping an item to the Buyer. To check that the transaction is valid before including it in a block, a miner checks one public key signature, and queries the index of unspent coins, which takes $O(logm)$ time overall, where m is the size of the index. The Seller does not need to fetch the entire contents of each block - it is sufficient only to check just the proof-of-work, since it is assumed that the hashpower of the honest participants on the network is steady, and only mine valid blocks - this is an example of SPV security. Note that the Buyer must trust the Seller to fulfill his end of the arrangement.
- (Escrow Purchase) If the Buyer does not wish to trust the Seller, but both parties can agree to trust a willing third party, then the third party can act as an Escrow agent. The Buyer constructs a transaction that allows the Escrow agent to choose whether to direct the funds to the Seller, or to the Buyer as a refund. The key idea is that the Escrow can only take these two options, and cannot take the funds for himself. Therefore the Escrow need not be fully trusted, but only must be trusted not to collude with either party. This model

of Escrow is used by the BitMit auction site, as well as the former black market site SilkRoad. Independent Escrow provided by a company called "BTCrow." The threat of damaged reputation may keep the Escrow honest. On the other hand, if there is a dispute, where both parties accuse the other of lying, the Escrow may essentially have no way to determine which is at fault.

- (Time-locked Refund) The Escrow may abort and not sign any message, which would result in the funds being frozen indefinitely. Bitcoin supports a timelock mechanism, that allows a refund transaction to be prepared in case of the escrow agent timing out.
- (Green Address [7]) If payment is received from a trusted party, then it may not be necessary to wait for confirmations. A Green Address refers to a publicly known address associated with a reputable entity, such as a hosted account provider (i.e., a bank). This is analogous to a cashier's check. If a double spend occurs, it would be easy to publish evidence of such, tarnishing the entity's reputation. MtGox has provided the option of withdrawing payments via a well-known Green address.
- (Quick Purchase) Bitcoin vending machine [8] Also several companies are currently deploying machines exchanging Bitcoin for other currencies, in particular Lamassu and BitcoinATM. The challenge is in accepting the coins in a short period of time. The approach is to connect to a large number of nodes and attempt to detect double spends, in which case can enter a panic mode.
- (Pay-to-Script-Hash) In the case of an escrow transaction, for example, the sender must create a transaction that includes the hashes of several public keys. The sender may have to pay a fee proportional to the size of the transaction. As a minor optimization, the sender can construct a transaction that includes just the hash of a larger script containing these public keys, and the full script must be provided by the receiver. This allows the sender to pay less in fees - the recipient may pay fewer fees overall to make the next transaction, if it is not time critical. [9]
- (Micropayments) Suppose a Client wants to have a per-minute subscription for some service provided by a Server, and to decide whether to continue at each minute. Publishing a new transaction each minute would require a lot of transaction fees and storage on the network. Instead, the client creates a transaction that "bails in" the maximum amount of money for the day, along with a refund transaction that times out at the end of the day. After each minute, if the client wishes to continue, he transmits a new transaction to the server that increases the amount. When the server is ready to claim the transaction, he signs the largest transaction and publishes it. Only one transaction ever needs to be published, yet the Server is never able to take more than the Client offered.
- (Balance query, auth index) So far, every transaction has involved only point-wise queries to the index. An example of a more complicated query would be to compute the total balance of an address, meaning the sum of all coins associated with a particular public key. This could be done using a linear scan of the unspent transaction outputs index, but this would be inefficient. Instead, a second index could be sorted according to public key (at least for standard transaction types). This would increase the storage cost to a validator by a constant factor. In any case, the user performing the query would be required to store the entire index as well. Alternately, suppose that the unspent transaction index is committed to as an authenticated data structure, the root of a merkle hash tree. Then the user can confirm the balance using only a $O(logm)$ size transmission from an *untrusted* node.

2) Overlay Currencies: Besides using Bitcoin as money, Bitcoin as a platform can also be used to maintain alternate currencies, such as personally-issued IOU coins, transferable shares in a Business, etc. A colored coin is introduced by fiat - an issuer, such as the executive of a corporation, declares that a particular quantity in the new protocol should be regarded as a currency. The protocols place restrictions on how these quantities are used in transactions, such as guaranteeing they are conserved.

- (MasterCoin [10]) MasterCoin proposes to create overlay currencies by building an entirely separate transaction protocol that uses Bitcoin as the underlying global append-only log. The observation is that arbitrary data packets can be embedded in Bitcoin transaction messages, and given a sequential ordering based on the contents of blocks. The first valid sequence of data packets is considered authoritative. The biggest drawback to this approach is cost; because Bitcoin is used only as an append-only log, miners do not perform any validation. The validity of a particular data packet cannot be determined except with knowledge of the previous history of transactions. Therefore determining whether a MasterCoin packet represents a valid coin requires processing every Bitcoin transaction in order to determine if it is a Mastercoin transaction, and maintaining a sufficient index.
- (Colored [11], [12] and SmartCoins [13]) Colored coins have similar goals to MasterCoin. The difference is that colored coins make use of the transaction structure of ordinary Bitcoin, rather than using a separate data protocol. Each unit of a colored coin is "carried" by units of Bitcoins. There is thus some corresponding notion of SPV security. It can only be valid if every input corresponds to a previous valid transaction, etc. However, in order to determine the color of the coin, you must traverse the entire relevant subgraph of transactions, bounded by $n_{\mathcal{C}}$, the number of transactions associated with color $\mathcal{C}$.
- (Freimarkets [14] and Ripple) Whereas Colored Coins and Master Coin involve the creation of overlay currencies using, Ripple proposes to build the rules governing such overlay currencies directly into the validation rules. This increases the cost of validation, however it restores the ability to use SPV security, where users do not need to store the entire index.

Note that Ripple also proposes a different consensus model based on designated trusted entities rather than incentivized proof-of-work.

3) Private Transactions: Standard Bitcoin transactions are published in the clear, although account numbers are typically not associated with any real world identity. Publishing transactions makes it easy to verify that system invariants are maintained, for example that the total amount of currency is conserved. However, many users would prefer their financial transactions to be kept private. There are several approaches to obscuring relations between keys, essentially by performing a mix of some kind. The simplest solution involves a third party server that is trusted to delete its logs. On the other hand, the following two approaches enable secure coin-mixing without the involvement of a third party.

- (CoinJoin) A shuffle of n transaction inputs and n transaction outputs can be conducted atomically using a single transaction. However, this requires several rounds after which each party signs the entire transaction. Notably, it is easy for an attacker to prevent the mix from completing.
- (ZeroCoin) [15] Perhaps the most thorough solution is to embed the functionality of a publicly verifiable (and public coin) third party mix directly into the validation rules of the block chain. This incurs additional validation cost.

4) Credentials:

- (Namecoin) An altcoin has been proposed to serve the functionality of the current Domain Name System. The validation rules are augmented to include updates of domain name records. Unused domain names cost a minimal fee to register initially, but the owner of a domain name can sell it (atomically) for an arbitrary price. Periodic renewal fees discourage squatting of domain names.
- (Fidelity Bond) A general approach to meaningful identities in an otherwise anonymous environment is to require something of value invested in each identity. By burning or encumbering a quantity of money associated with a particular name, a party can signal that they value the reputation associated with this identity.
- (Anonymous Credentials) The **Issuer** code here indicates that the root identity issuer must be trusted to assign identities correctly.

5) Cross-Chain Transactions: Bitcoin is not the sole cryptocurrency, instead it must be understood as the currently-dominant member of a potentially unbounded ecosystem of possibly competing or cooperating crypto-currencies. The simplest interaction between two related currencies is akin to a foreign currency exchange.

- (TierNolan's Protocol [16]) There exists a three phase process by which two currencies can be atomically exchanged
- (P2PTradeX [17]) Another proposal involves having one currency validate blocks in the other. This potentially allows for more complicated interactions, as validators of one currency essentially enjoy a reduced-cost SPV view of the other chain, and vice versa. (example needed)

6) Other:

- (Coin Flip Lotteries) Using Bitcoin has a clear appeal to online gambling sites (including illicit ones), since it allows users to cash in and out without having to interact with ordinary banking institutions. While several gambling sites feature games like poker[AM: cite here] or elaborate virtual casinos[AM: cite minecraft bitcoin casino], some of the most popular have been very simple roulette games, such as Satoshi-dice [AM: cite satoshi dice]. [AM: TODO: add a comment about the trend of "provable fairness" in Bitcoin gambling, and the way that Satoshi-dice uses Bitcoin transactions]. Multi-player lotteries of this sort can also be conducted directly on top of the Bitcoin blockchain protocol, without any counterparty risk. [18]
- (CommitCoin) In CommitCoin, the hash of a document is embedded in the public key of a transaction that is published and timestamped in the blockchain. The user must store the document herself, and at any later time can publish the document and prove to anyone else that the hash of the document was known previously to the timestamped date. [19]

TABLE I. EVALUATION SUMMARY OF PROTOCOLS USING BITCOIN AS A CURRENCY AND/OR A TRANSACTIONAL PLATFORM (m IS THE NUMBER OF ELEMENTS IN THE INDEX, n IS THE NUMBER OF BLOCKS FOR THE DURATION OF THE PROTOCOL, c IS THE NUMBER OF TRANSACTIONS IN A BLOCK

	Validation Cost Per Tx [Overall]			Party Cost per Tx [Overall]							
Protocol	**Comms**	**CPU**	**Space**	**Comms**	**CPU**	**Space**	**Time**	**Trust**	**Fork**	**Advantages**	**Disadvantages**
Purchases											
Standard	1	$\log m$	1	1 $[nc]$	$\log m$ $[nc \log m]$	1 $[m]$	1K	Merc	No		
SPV-security	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	1K	Merc	No	Less party cost	
Escrow (1 of t sigs)	t	$t + \log m$	t	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	2K	3rd	No	No Merc trust	Limited 3rd party
Green Address	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	1	Merc,3rd	No	Instant confirm	3rd party trust
Micropayments	$1/f$	$1/f \log m$	$1/f$	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	2K	None	No	Low cost, trust	Incr. goods
Overlay Coins											
MasterCoin	1	$\log m$	1	1 $[nc]$	$\log m$ $[nc_{\mathcal{M}} \log m_{\mathcal{M}}]$	1 $[m_{\mathcal{M}}]$	1K	Issuer	No		High overhead
Colored/Smart Coins	1	$\log m$	1	$nc + \log c$ $[n]$	$nc + \log c$ $[n]$	1 [1]	1K	Issuer	Lower overhead		
Freimarkets	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	1K	Issuer	Yes	Low overhead	Fork required
Private Transactions											
Zerocoin	1	$\log m + \log n$	1 $[m + n]$	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	2K+ w	None	Yes		High cost
CoinJoin	q	$q \log m$	q	$q + \log c$ $[n]$	$q + \log c$ $[n]$	q [1]	1K	None	No		DoS
Credentials											
Namecoin	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	1K	None	Yes		
Fidelity Bond	1	$\log m$	1	$n + \log c$ $[n]$	$n + \log c$ $[n]$	1 [1]	1K	None	No		Expensive
Anonymous Credentials	1	$\log m$	1	$\log c$ $[n]$	$nc + \log c$ $[n]$	1 [1]	1K	PKI	No		Scanning cost
Other											
CommitCoin	1	$\log m$	1	$d + n + \log c$ $[n]$	$n + \log c$ $[n]$	d [1]	1K	None	No		
Coin Flip	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	2K	Pub. Verif.	No		
Transaction Puzzle	1	$\log m$	1	1 [1]	w [1]	1 [1]	$w + 1/\alpha$	None	No		Miner only
Cross-Chain Swap											
TierNolan's protocol	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	3K	None	No		
P2PTradeX	1	$\log m$	1	$\log c$ $[n]$	$\log c$ $[n]$	1 [1]	2K	None	Alt		Altcoin only

VI. Proposed Modifications to the Platform

Many proposed modifications to Bitcoin are meaningful independent of any particular protocol. Generally these improve security, decrease the cost of participation, or increase the speed at which transactions can be included. Often there is a tradeoff, for example increased security at the cost of. Many involve changing parameters which are configured in Bitcoin without any particular justification.

A. Description of the Columns

Proposed modifications to the Bitcoin platform can be evaluated according to the following dimensions.

- (System efficiency) Does it decrease the cost (storage, computation, communication) of transaction or block validation?
- (Economic structure) Besides efficiency, does it alter the reward for miners to participate?
- (User Centric) For users of the system (participating in protocols involving transactions or queries to the system) does it increase their security? Or reduce their cost of participation?
- (Security Model) Does it reflect a change or refinement of the participation or overall attack model?
- (Integration Requirement) Does it require a change to the block validation rules (hard fork), or the transaction inclusion rules (soft fork), the peer-to-peer messaging protocol, or just the client?
    - (Hard-Fork) For new functionality that existing full-validating clients (whether or not they also mine) will reject. If miners switch to this, cause disagreement between full nodes and SPV nodes. Hard-fork requires every validating node to upgrade their clients immediately.
    - (Soft-Fork) Refinements of existing behavior, for example disabling an opcode, enforcing a subset of the transaction script language. A majority of mining participants can enforce a soft-fork by refusing to work on blocks violating the new rule.
    - (Miner Policy) Miners can choose not to include transactions into their own blocks, yet still accept blocks made by other miners. IsStandard and eligius. Transaction prioritization.
    - (P2P) Affects the messaging layer. How nodes find each other, announce, request, transfer data such as blocks and transactions. How to serve old blocks to new nodes on the network.
    - (Client) Only requires a change of the client.
- (Maturity): is there an implementation available? Has it been implemented in an altcoin? Or has it been adopted as a patch to Bitcoin? Any particular change can be made as an altcoin.

B. Description of the Rows

1) Hard-Fork Parameters: The outermost layer of the system.

- (Difficulty adjustment) Time is a critical parameter. If this is set too low, then the network does not reach consensus, or the effective power of the network against an attacker is diminished. If it is set too high, then the variance for measuring the amount of security is lower. This is set by magic parameter. How else should it be set?
- (Alternate Mining Systems) Alternate functions based on hash function, such as scrypt. Useful proof-of-work. Proof-of-stake.
    - (Proof-of-Stake) Alternate security model. Perceived possibility of an economic attack.
    - (Designated authorities) Trusted (or semi-trusted) servers may be designated to receive, publish, and linearize transactions [20]. Another point out is Ripple's approach to unique node lists. We do not understand the model under which these are secure. On the other hand, we don't yet have a clear model in which Bitcoin's incentives work either.
    - (Useful proof of work) Bitcoin appears to be wasteful. There are two ways in which it is wasteful - one is that the energy burned is not directly useful. A common suggestion is to replace the proof of work with a search function, such that a solution would be useful, distributed search project. It has been argued (e.g., by Kroll et al) that this must be a pure public good, or else it subsidizes an attacker. Another form of waste is that the investment in more efficient mining devices has led to the development and distribution of devices that are useful for computing SHA2 and nothing else. Permacoin: in concurrent work we investigate the potential to replace the puzzle with a "storage hard" puzzle, so that mining equipment is copies of a dataset. [21]
- (Block Size Limit) Remove it all together? [AM: Should summarize Petertodd's arguments here]
- (Monetary Policy) Bonus schedule. Demurrage. Coinbase maturity. Conservation invariant.
- (Transaction Scripts) New op codes, including ones disabled. Other signature schemes. Access to the index? Suspended computation. Ability to refer to other scripts. Turing complete? Succinct proofs with TinyRAM/Pinocchio? Access to additional indexes?

Not necessarily have to be linear to be useful, although some conflicts such as a double-spend require linearization.

2) Mining Behavior: Although miners are bound in many ways by the agreed-upon rules — no miner can commit an invalid block, for example, since other miners will reject it — miners are also given a lot of discretion, especially in the form of choosing which transactions to include in a block, and which transactions to relay. In particular, a miner can choose to ignore a transaction, and it's difficult for other miners to punish them — perhaps the miner legitimately never received the transaction in the first place. On the other hand, miners tend to follow common conventions (known as "miner policies" or

"soft-fork rules") and can in general be expected to respond to rational incentives. The simplest example of miner policy is prioritizing transactions that carry fees. Proposed variations of miner policies are enumerated as follows:

- (Transaction Preference) Priority. Fees. Dust discouragement. Refusing to mine on non-standard transactions (policy change). For example, the Eligius mining pool mines nonstandard transactions. Probabilistic transaction validation.
- (Block Preference) No standard client has proposed anything other than take the longest block, and most recent one found if there is a tie. Kroll et al. argue that taking the longest block is a focal point. Miners may decide to refuse mining on blocks containing transactions they don't like. If a miner deviates from the consensus here, then it will waste its mining power. If enough take it, then it is a "soft-fork rule change," can be used to enforce a subset of the rules.
- (Computationally Secure Verification) Commitment of index. Cryptographic improvements for validation. Efficiency Pinocchio or TinyRAM. Zero knowledge validation of commitments? ZeroCoin is a limited form of this. Compressed work sample.
- (Pool participation) Forms of rewards. P2Pool, pool hopping (cite meni rosenfeld), reward schemes. Apparent desire for low variance. Stratum mining protocol and getwork.

3) P2P System: The system that passes around blocks and transactions. Primarily concerned with denial of service, while maintaining connectivity. Also concerned with privacy. Conceptually the entire P2P network could change, or a new P2P network could be developed, that interfaces with other Bitcoin nodes.

- (Relay Transaction/Block Preferences) Similar to mining behavior, although applies to other nodes not just miners.
- (Header only block download) [AM: TODO: summarize] [22].
- (Faster block relaying) Fast block propagation is crucial for the network to converge on recent blocks. Christian Decker et al. have studied block propagation and proposed faster variants [23]; there is also an earlier implementation by Luke-Jr[8]. Greg Maxwell proposed network coding as a method to reduce the cost of transmitting blocks, assuming that a peer already knows some portion of the block. [24]
- (Client puzzles) Denial of Service is a common problem in networks, although fairly few of the known countermeasures are currently implemented in Bitcoin. Bitcoin nodes are essentially open to strangers, and are therefore vulnerable to resource exhaustion attacks. One countermeasure is to have a potential client solve a small proof-of-work before allocating resources. Ordinarily, such puzzles represent unrecoverable waste; on the other hand, since the Bitcoin network already makes use of puzzles, a natural approach is to ask potential clients to perform a small amount of Bitcoin mining work before allocating resources, thereby getting the benefit with no additional waste — this general approach is known as a bread-pudding protocol. [25]

4) Client/Agent:

- (SPV vs Full Node Security)
- (Wallet, key storage) Encrypted wallet, HD wallet, brain wallet, paper wallet, etc.
- (Transaction Construction) Making change. This is where transaction anonymity is attacked, linking keys. Change transaction.
- (Hardware clients) Although most end users interface with Bitcoin using clients for general purpose computers or mobile devices, storing private keys on specialized hardware such as smart cards or secure tokens have several potential advantages. These devices may be made tamper-proof, can be kept in the user's (physical) wallet or keychain, aren't vulnerable to computer viruses affecting general purpose operating systems, need not connect directly to the internet, and may even present a more convenient user interface. Bitcoin hardware wallets include the TREZOR[9]. Other specialized hardware Bitcoin clients include vending machines [8] or ATMs [10].

VII. Discussion

A. Smart Contracts a Universal Platform

- Older documents from the cypherpunk era about smart contracts as a general abstraction

– Mark Miller's thesis, and "Ode to the Granovetter" paper about capability-based financial contracts [26]

– Nick Szabo's papers on smart contracts, coining the word. [27]

- Current attempts towards smart contracts, including Ethereum, OP_EVAL, and Mike Hearn's wiki page about bonds and financial contracts [28].

B. Interactions between Bitcoin and competitors

One of the murkiest areas is understanding Bitcoin's security in the context of the larger ecosystem of similar cryptocurrencies, where multiple separate and concurrent protocols compete for the same computing resources. Bitcoin's security model essentially requires an amount of mining participation larger than any attacker. Bitcoin's incentive mechanism functions similarly to a fundraiser or a recruitment drive; the more participation, the more an attacker would have to pay to defeat it. This approach to security could be described as safety in numbers. However an implication is that for Bitcoin to be

[8]Source code commit by Luke-Jr,https://gitorious.org/bitcoin/luke-jr-bitcoin/commit/4e54ea804ccdd2223e622497f0d46cceb27b9d22

[9]TREZOR product website, http://www.bitcointrezor.com/

[10]See for example Robocoin, Genesis1, and Lamassu[AM: cites needed here]

secure, it must stay on top, and attract the greatest amount of participation. [11]

C. Bitcoin as money

Three definitions are normally given as money. 1) Unit of account, 2) means of exchange, 3) store of value. Do these form design criteria for a system like Bitcoin? In other words, do these requirements form a problem statement to which a system like Bitcoin can be derived as a solution? Empirically, Bitcoin satisfies these. It has maintained a value over three years, it can be used bin exchanges, and accounts are denominated in BTC. However it isn't clear how we can argue that Bitcoin will maintain this. Are the other coins money as well?

The idea that money comes from barter can be interpreted as a matter of theory, rather than a historical account. Money is *reducible* to barter, in the sense that given the ability to exchange items of value (i.e., commodity goods), one of the goods naturally becomes used as "money". This is the approach taken in the Kiyotaki-Wright model of money. However, the other direction does not seem to hold - to the extent Bitcoin implements money, money is not sufficient to enable a barter. Trading virtual currency for mail order goods, seems to require external mechanism of some kind, such as a trusted mediator, legal enforcement, or the threat of a tarnished reputation.

Through the lens of Bitcoin and the associated ecosystem of cryptocurrencies, we may have an unexpected opportunity to observe the emergent formation of money (or something similar) under different environment conditions.

D. Remaining Open Questions

How to theoretically model Bitcoin? Especially in a way that accounts for an ecosystem with multiple currencies cooperating or competing for participation.

How to match fees to costs?

How to have scalable participation, where the state is not replicated to the network in its entirety, but instead load-balanced in some way?

VIII. Conclusion

References

[1] T. B. Lee, "Major glitch in Bitcoin network sparks sell-off; price temporarily falls 23%," http://arstechnica.com/business/2013/03/major-glitch-in-bitcoin-network-sparks-sell-off-price-temporarily-falls-23/.

[2] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proceedings of Crypto*, vol. 82, 1982, pp. 199–203.

[3] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social Networks*. Springer, 2013, pp. 197–223.

[4] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *Financial Cryptography and Data Security*, March 2014.

[5] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of Bitcoin mining, or Bitcoin in the presence of adversaries," in *Workshop on the Economics of Information Security*, Jun. 2013.

[6] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, March 2014.

[7] jav, "Instawallet introduces new approach to instant payment: Green address technique," https://bitcointalk.org/index.php?topic=32818.0, July 2011.

[8] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten, "Have a snack, pay with bitcoins," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*. IEEE, 2013, pp. 1–5.

[9] G. Andresen, "Pay to script hash," Bitcoin Improvement Proposal (BIP16) https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki, 1 2012.

[10] J. R. Willett, "Mastercoin complete specification, v1.1," https://sites.google.com/site/2ndbtcwpaper/MasterCoin%20Specification%201.1.pdf, 2013.

[11] M. Rosenfeld, "Overview of colored coins," 2012. [Online]. Available: https://bitcoil.co.il/BitcoinX.pdf

[12] killerstorm, "Armoryx (colored coins): issue and trade private currencies/stocks/bonds/etc," 2012. [Online]. Available: https://bitcointalk.org/index.php?topic=106373.0

[13] J. Garzik, "Smartcoin: Distributed smart property software," 2012. [Online]. Available: https://github.com/jgarzik/smartcoin

[14] M. Friedenbach and J. Timon, "Freimarkets: extending bitcoin protocol with user-specied bearer instruments, peer-to-peer exchange, off-chain accounting, auctions, derivatives and transitive transactions," http://freico.in/docs/freimarkets.pdf, August 2013.

[15] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," *IEEE Symposium on Security and Privacy*, 2013.

[16] T. Nolan, "Alt chains and atomic transfers," May 2013. [Online]. Available: https://bitcointalk.org/index.php?topic=193281#msg2224949

[17] S. D. Lerner, "P2ptradex: P2p trading between cryptocurrencies," https://bitcointalk.org/index.php?topic=91843.0, July 2012.

[18] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin."

[19] J. Clark and A. Essex, "Commitcoin: carbon dating commitments with bitcoin," in *Financial Cryptography and Data Security*. Springer, 2012, pp. 390–398.

[20] B. Laurie, "An efficient distributed currency," Online, =http://www.links.org/files/distributed-currency.pdf, 2011.

[21] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in *To appear, IEEE Security and Privacy (Oakland)*, May 2014.

[22] P. Wuille, "Switch to headers-based synchronization," Pull request, https://github.com/sipa/bitcoin/commit/b230a6599e884e158fe49e2cf946801eea83dcaa.

[23] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*. IEEE, 2013, pp. 1–10.

[24] G. Maxwell, "block network coding," Wiki article: https://en.bitcoin.it/wiki/User:Gmaxwell/block_network_coding.

[25] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks*. Springer, 1999, pp. 258–272.

[26] M. S. Miller, C. Morningstar, and B. Frantz, "Capability-based financial instruments," in *Financial Cryptography*. Springer, 2001, pp. 349–378.

[27] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[28] M. Hearn, "Distributed bond markets and pay-to-policy outputs," 2012. [Online]. Available: https://bitcointalk.org/index.php?topic=92421.0

[11] For example, in 2012 a Bitcoin mining pool attacked a small alt-coin, CoiledCoin. A minor participant on the larger Bitcoin network could easily join a smaller network and overwhelm it.