MultiChain Private Blockchain – White Paper

Dr Gideon Greenspan, Founder and CEO, Coin Sciences Ltd

Background

Bitcoin is now recognized as a cheap, rapid and reliable method for moving economic value across the Internet in a peer-to-peer manner. Aside from a brief fork between incompatible versions in March 2013, the bitcoin network has been operating continuously and smoothly for over 5 years. Although there have been losses and thefts of bitcoins belonging to individual holders, the network itself has never been successfully attacked or impeded.

Despite bitcoin's many technical achievements, it is far from reaching mainstream consumer or business adoption. Judging by the current trend in transaction volumes[1], the sluggish growth of bitcoin usage shows no sign of changing in the foreseeable future. This is despite the availability of many easy-to-use bitcoin wallets and the fact that bitcoin can now be spent online at many mainstream businesses such as Microsoft, Dell and Overstock.

There are many possible causes of bitcoin's slow adoption, including: (a) end-user satisfaction with existing payment systems, (b) the practical difficulty of purchasing bitcoins, (c) the volatility of bitcoin's value relative to government-issued currencies, (d) the perception that bitcoin is insecure, (e) questions over bitcoin's legal status, (f) the irreversible and unforgiving nature of bitcoin transactions, and (g) a lack of support for bitcoin in the mainstream financial sector.

In the absence of end-user adoption, many have suggested that bitcoin could help improve internal processes within the traditional financial sector, by lowering costs, reducing settlement times and eliminating intermediaries. One immediate theoretical possibility is using bitcoin as a currency and conduit for rapid inter-bank settlement. However the volatility of bitcoin's value relative to government-issued currencies renders this unworkable in practice. A more promising direction is to use bitcoin's infrastructure to transact in assets other than bitcoin itself.

Blockchains and tokenization

At the heart of bitcoin lies the blockchain, a global decentralized ledger which stores the full history of all bitcoin transactions. The blockchain is verified and stored by every node in the bitcoin network, of which there are approximately 6,000 in June 2015[2]. The bitcoin protocol ensures that, barring temporary discrepancies, every node in the network has the same version of the blockchain, without requiring this consensus to be determined by a central authority. Another key feature of bitcoin is that nodes can join or leave the network at any time, without disrupting the functioning of other nodes or the ongoing processing of transactions.

New transactions can be created by any node and are propagated across the network in a peer-to-peer fashion. Any node can take a set of these pending transactions and create ("mine") a new block containing them together with a link to the previous block. The new block "confirms" the transactions and is also propagated across the network. To prevent minority control over mining, bitcoin uses "proof of work" to make it computationally difficult and expensive to create a new block. If a "fork" occurs, in which two competing blocks are mined almost simultaneously, proof of work

[1] Source: https://blockchain.info/charts/n-transactions?timespan=all&daysAverageString=7

[2] Source: https://getaddr.bitnodes.io

also acts as a dispute resolution mechanism. Since blocks are hard to create, it is unlikely that both forks will grow at an identical speed. The protocol specifies that the fork with the greater amount of work is the correct one, so the network quickly regains a unified global consensus.

Along with bitcoin transactions, the blockchain can be used to store any digital data. While some view such uses as "bloating the blockchain", bitcoin's decentralized nature means that they cannot effectively be stopped. This led the developers of Bitcoin Core, the official bitcoin client, to introduce an official mechanism for adding arbitrary metadata to transactions in early 2014[3]. This mechanism is used by services such as Proof of Existence and BlockSign to notarize the existence of a document by embedding a digital signature of that document inside a transaction. Other tools such as php-OP_RETURN enable larger pieces of data to be stored and retrieved from the blockchain, turning it into a general-purpose permanent decentralized data store.

Transaction metadata is used by several protocols, such as CoinSpark, Counterparty, Omni Layer and Open Assets, to support third party assets on the bitcoin blockchain. First, an issuing entity creates a new set of tokens representing an asset, by sending a transaction with some "asset genesis" metadata. As part of this process, the issuer can undertake a contractual obligation to allow these tokens to be exchanged for the equivalent real-world asset at any time[4]. Ownership of the tokens is freely transferred between holders using other transactions with "transfer" metadata, without requiring the approval of the issuer or any other authority. In effect, a token acts as a digital bearer bond, with the ownership of that bond determined by the data embedded in the bitcoin blockchain[5]. In the finance world, a token issued by an institution with a strong credit rating could be perceived by other institutions as a close approximation to the underlying asset.

Bitcoin's shortcomings

Notwithstanding the promise of asset tokenization protocols, there are several reasons why the bitcoin blockchain is not yet suitable for institutional financial transactions. The problems can be divided into two groups, the first of which relates to scalability and cost:

- Limited capacity. The bitcoin blockchain currently supports around 300,000 transactions per day, as determined by its maximum block size of 1MB[6]. This capacity must be shared between all network users and is clearly insufficient for many financial applications. For example, the Visa network currently handles 150 million transactions per day in the USA[7]. While the maximum block size will likely increase in future, there is an ongoing debate over how quickly this can happen without driving out regular users and increasing the frequency of forks in network consensus[8]. In any event, institutional users cannot control the pace of this change, which will ultimately be determined by miner adoption.

---

[3] The mechanism is `OP_RETURN` outputs. See: http://blog.bitcoinfoundation.org/core-development-update-5/ for discussion or http://coinsecrets.org/ for examples of how `OP_RETURN`s are being used. Metadata can also be added in less network-friendly ways, e.g. by using fake addresses in multi-signature outputs.

[4] Example tokenization asset contract: http://coinspark.org/create-asset/?file=contract&id=sample&template=uk

[5] Regulation has made bearer bonds almost extinct in the US, due to their convenience for money laundering and tax evasion. Blockchains improve on traditional bearer bonds by storing a full audit trail of all transactions.

[6] Analysis of maximum transaction rates: http://hashingit.com/analysis/33-7-transactions-per-second

[7] Source: http://usa.visa.com/merchants/industry-solutions/retail-visa-acceptance.jsp

[8] See http://www.ofnumbers.com/2015/02/06/what-is-the-blockchain-hard-fork-missile-crisis/ and https://letstalkbitcoin.com/blog/post/lets-talk-bitcoin-217-the-bitcoin-block-size-discussion for discussion.

- Transaction costs. The standard fee per bitcoin transaction is currently BTC 0.0001 (2.5 cents at $250/bitcoin) and is collected by the miner of the block in which that transaction is confirmed. While this fee is optional, transactions with lower fees can encounter significant delays in confirmation. This sum is already a non-trivial tax on transactions of small monetary value. Furthermore, when the demand for bitcoin transactions outgrows the supply of available block space (see previous point), this fee may increase substantially, as transactions are forced to bid with each other to compete for inclusion in a block.
- Irrelevant data. Institutions deploying over the bitcoin network need to process and store a large quantity of information that is of no interest to them. When a new bitcoin node is launched, it first downloads, verifies and stores the entire history of all bitcoin transactions. Going forwards, it must also verify all new transactions and blocks created, even though most are of no relevance to the user of that node. This problem is avoided by lightweight nodes, which can transact over the blockchain without storing it. However, their weaker security renders them unsuitable for serving as the backbone of an institutional system.

The second group of problems relates to privacy and security:

- Mining risks. Bitcoin's proof of work mining is an open global race to solve the difficult mathematical problem required to create a new block. While this process is well suited for a general purpose decentralized network, it entails several risks for institutional users: (a) the unpredictable delay for transaction confirmations, with block creation times defined by a Poisson distribution with average 10 minutes, (b) the risk of some miners refusing to confirm institutional transactions for ideological or economic reasons, (c) the potential for a 51% attack, where a group of miners controlling over half of the network's computational power collude to rewrite a significant period of the blockchain's recent history[9]. Although such an attack is unlikely to occur, it clashes with the institutional need for transactions to be absolutely irreversible once settlement has taken place.
- Lack of privacy. By design, all bitcoin transactions are visible to all network nodes and therefore to the entire world via blockchain explorers such as blockchain.info. This public aspect is mitigated by the fact that bitcoin addresses cannot easily be connected to their real-world owners. Nonetheless the existence and rate of transactions cannot be hidden, and participants run the risk of their identities being revealed at some point in future, at which point their entire transaction history could be retroactively inferred[10].
- Openness. Anybody with an Internet connection is able to connect to the bitcoin network and transact with other participants. This makes bitcoin an attractive conduit for illegal transactions, since Know Your Customer (KYC) checks cannot be enforced at the network level. While it is certainly possible for regulated institutions to ensure that they (or their customers) only transact with known counterparties, this requires every transaction to be individually vetted, creating a significant burden in terms of architecture and workflow.

Can financial institutions enjoy the benefits of blockchains without suffering from these problems?

---

[9] Apart from the well-known 51% attack, some researchers argue that 33% of the network capacity would be sufficient: http://arxiv.org/abs/1311.0243. Either way, tokenized assets break the fixed relationship between the cost and potential reward of conducting such an attack, since the reward can contain more than just bitcoin.

[10] See https://en.bitcoin.it/wiki/Anonymity and http://arxiv.org/abs/1107.4524 for discussion.

Other blockchains

To begin answering this question, we can point out that bitcoin is far from the only public blockchain in active operation. Hundreds of other blockchains have been created, each of which has its own cryptocurrency, network of nodes and rules for generating consensus. Some examples include Litecoin, BitShares, Nxt, Dogecoin and Namecoin[11]. Many but not all of these blockchains run on software which is derived from bitcoin's source code, with only minor modifications to mining algorithms or other parameters.

Despite this proliferation of innovation (and, in many cases, pump-and-dump schemes), bitcoin maintains its position as the predominant cryptocurrency. The obvious cause is the large number of individuals and businesses who already hold or accept the bitcoin currency. However there is also a significant network effect in terms of mining. Bitcoin is the cryptocurrency backed by the most mining power, making it the most secure against 51% attacks. This gives it the highest perceived value and therefore the highest market capitalization. Consequently bitcoin offers the highest financial reward to miners and this leads it to attract and retain more miners than any other currency, increasing its value further. This virtuous loop will be hard for any other blockchain to beat unless it provides some crucial new functionality.

The bitcoin blockchain uses a per-output transactional model, in which every transaction has a set of inputs and a set of outputs. Each input "spends" one output of a previous transaction, with the blockchain ensuring that this output cannot be spent elsewhere. The full history of transactions forms a multi-way connected chain[12], which terminates at the "coinbase" transactions in which miners are awarded new units of the currency. All of the bitcoin in a transaction's inputs flow into that transaction, which are then distributed across its outputs in accordance with the quantities written within. As a result most regular payments require two outputs – one with the intended amount for the recipient, and the other containing "change" which goes back to the sender for use in a subsequent transaction. A transaction is only valid if it has sufficient total bitcoin in its inputs to cover the total written in its outputs, with the difference forming the miner's fee.

Some blockchains, such as Nxt and Ethereum, use a simpler per-address transaction model, where transactions have no inputs and outputs per se. Rather, each transaction moves funds from one address to another, without indicating the specific previous transaction from which those funds should be taken. Transactions are only valid if the sending account has sufficient balance in order to make the payment specified. This model has both advantages and disadvantages compared to bitcoin's per-output model[13]. We favor the per-output model in order to maintain compatibility with bitcoin and for its suitability for highly concurrent parallel processing.

All of these blockchains are open, allowing anyone on the Internet to connect, transact or mine. However this is not a necessary characteristic. To begin with, one can imagine a small group of entities coming together to create their own bitcoin-like blockchain, with access restricted to specific IP addresses[14]. Members would transact with each other in a closed but decentralized network, perhaps using tokenized assets. However, the problem with a small, closed network based on proof

---

[11] See http://coinmarketcap.com/all/views/all/ for an exhaustive list.

[12] The formal name for this in Computer Science is a directed acyclic graph (DAG).

[13] See https://github.com/ethereum/wiki/wiki/Design-Rationale#accounts-and-not-utxos for discussion.

[14] Indeed, Bitcoin Core already contains a `connect` setting to enable such a restriction.

of work is that one aggressive participant can easily take over the mining process by purchasing specialized hardware. This would give them up to a million times more mining power than other members using regular computer processors[15]. As the miner of every block, they could unilaterally decide which transactions get confirmed or rewrite the blockchain's history at any time. Furthermore, they could avoid detection while doing this, by using a different public address to collect the mining rewards for each block. It is crucial for any solution for private blockchains to ensure that one or a minority of participants cannot seize control in this manner.

Introducing MultiChain

MultiChain is an off-the-shelf platform for the creation and deployment of private blockchains, either within or between organizations. It aims to overcome a key obstacle to the deployment of blockchain technology in the institutional financial sector, by providing the privacy and control required in an easy-to-use package. Like the Bitcoin Core software from which it is derived, MultiChain supports Windows, Linux and Mac servers and provides a simple API and command-line interface. In the next few sections we describe the first public release of MultiChain. Later we discuss several other features on the MultiChain roadmap.

Private blockchains

MultiChain solves the related problems of mining, privacy and openness via integrated management of user permissions. The core aim is threefold: (a) to ensure that the blockchain's activity is only visible to chosen participants, (b) to introduce controls over which transactions are permitted, and (c) to enable mining to take place securely without proof of work and its associated costs. Once a blockchain is private, problems relating to scale are easily resolved, since the chain's participants can control the maximum block size. In addition, as a closed system, the blockchain will only contain transactions which are of interest to those participants.

To understand permissions in MultiChain, we begin by noting that all cryptocurrencies manage identity and security using public key cryptography. Users randomly generate their own private keys and never reveal them to other participants. Each private key has a mathematically related public address which represents an identity for receiving funds. Once sent to a public address, those funds can only be spent using the corresponding private key to "sign" a new transaction. In this sense, access to a private key is equivalent to ownership of any funds which it protects.

Beyond controlling access to funds, this type of cryptography enables any message to be signed by a user to prove that they own the private key corresponding to a particular address. MultiChain uses this property to restrict blockchain access to a list of permitted users, by expanding the "handshaking" process that occurs when two blockchain nodes connect:

1. Each node presents its identity as a public address on the permitted list.
2. Each node verifies that the other's address is on its own version of the permitted list.
3. Each node sends a challenge message to the other party.
4. Each node sends back a signature of the challenge message, proving their ownership of the private key corresponding to the public address they presented.

If either node is not satisfied with the results, it aborts the peer-to-peer connection.

---

[15] See https://en.bitcoin.it/wiki/Mining_hardware_comparison and https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison.

The principle of connecting permissions to public addresses can be extended to many other operations on the network. For example, the right to send and/or receive transactions can be restricted to a given list of addresses, since transactions reveal the addresses of both senders and recipients. Since transactions can have multiple senders and recipients, a transaction is only allowed if all of its senders and recipients are permitted[16]. Indeed in some cases we may prefer the blockchain to be fully publicly viewable and only apply restrictions on the ability to transact. Finally, by adding a signature field to the coinbase transaction included by miners in blocks, mining in MultiChain can similarly be restricted[17]. As discussed in the next section, this is crucial to preventing minority rule in a private blockchain.

In MultiChain, all privileges are granted and revoked using network transactions containing special metadata. The miner of the first "genesis" block automatically receives all privileges, including administrator rights to manage the privileges of other users. This administrator grants privileges to other users in transactions whose outputs contain those users' addresses together with metadata denoting the privileges conferred. When changing the administration and mining privileges of other users, an additional constraint is introduced, in which a minimum proportion of the existing administrators must vote to make a change. These votes are registered by each administrator in a separate transaction, with the change applied once sufficient consensus is reached[18]. The first few blocks of a chain define a "setup phase", in which a single administrator is able to bypass this voting process. Future versions of MultiChain could also introduce "super administrators" who can assign and revoke privileges on their own.

Since modifications to privileges are embedded in the metadata of transactions, they propagate quickly to all nodes in the network, creating consensus regarding the current state of play. However, because the network is decentralized, different nodes may receive permissions transactions at different times, either before or after other transactions. If the validity of a payment transaction depends on a privilege change that was broadcast shortly before, the difference could be critical, with some nodes accepting the payment and others rejecting it.

Any such differences will be resolved once the transactions are confirmed on the blockchain, fixing their final ordering. Every node follows the rule that transactions are 'replayed' in blockchain order, so each transaction in a block must be valid according to the state of user permissions immediately preceding it. If a transaction in a block is disallowed according to this rule, the entire block is rendered invalid. The miner of a valid block must also be on the permitted list after applying all privilege changes defined within that block's transactions[19].

---

[16] This rule is complicated by pay-to-script-hash (P2SH) outputs, in which recipients' addresses are only revealed when the output is spent. MultiChain offers several options: (a) allow any recipient and restrict by sender only, (b) only allow explicitly permitted P2SH outputs, (c) disable P2SH altogether. For multisig spends that use several addresses, MultiChain defines an input as permitted if at least one of the signers has the required permission.

[17] Note that a MultiChain wallet may contain multiple private keys and addresses. For each operation, MultiChain automatically uses a key (and corresponding transaction output) which is permitted to perform that operation.

[18] This voting process is complicated by the fact that an administrator can send multiple conflicting transactions regarding another user's privileges, and we need a reliable way of identifying their most recent vote. MultiChain solves this by including a monotonically-increasing timestamp in the metadata of each voting transaction.

[19] By checking mining permissions *after* applying the transactions in a block, we give administrators the emergency option of unfreezing a chain by immediately revoking miners who have gone quiet.

One privilege that falls outside this system is permission to connect, since it is not related to the blockchain's content. Instead, if this permission is revoked for a particular address, nodes immediately disconnect other nodes who used that address during handshaking.

For increased administrative convenience, temporary privileges can be granted by restricting them to a fixed range of block numbers. Transactions which depend on such privileges are only valid in blocks whose numbers are in the assigned range. For permissions changes that require consensus by voting, we only consider agreement to be reached if sufficient administrators have chosen the exact same block range for a particular user and privilege. This increases transparency for the network and relieves the administrative burden of remembering to revoke temporary privileges after their time has expired.

In order for a blockchain to be genuinely "private", for every address granted a permission on that chain, at least one administrator must know the real-world identity of the entity using that address. However most participants on the chain need not know each other's identities. A key feature of blockchains is allowing peer-to-peer exchange transactions, for example to swap tokens representing two different asset types. If addresses are kept anonymous, these exchanges can be performed without either party knowing the identity of its counterparty. One could imagine financial institutions transacting under many different addresses, with only regulators knowing which address belongs to which.

Mining in MultiChain

By restricting mining to a set of identifiable entities, MultiChain resolves the dilemma posed by private blockchains, in which one participant can monopolize the mining process. The solution lies in a constraint on the number of blocks which may be created by the same miner within a given window. MultiChain implements this scheme using a parameter called *mining diversity*, which is constrained by $0 \le$ *mining diversity* ≤ 1. The validity of a block is verified as follows:

1. Apply all the permissions changes defined by transactions in the block in order.
2. Count the number of permitted *miners* who are defined after applying those changes.
3. Multiply *miners* by *mining diversity*, rounding up to get *spacing*.
4. If the miner of this block mined one of the previous *spacing-1* blocks, the block is invalid.

This enforces a round-robin schedule, in which the permitted miners must create blocks in rotation in order to generate a valid blockchain. The *mining diversity* parameter defines the strictness of the scheme, i.e. the proportion of permitted miners who would need to collude in order to undermine the network. A value of 1 ensures that every permitted miner is included in the rotation, whereas 0 represents no restriction at all. In general, higher values are safer, but a value too close to 1 can cause the blockchain to freeze up if some miners become inactive[20]. We suggest a value of 0.75 as a reasonable compromise[21]. To conserve resources, nodes will not attempt to mine on a chain in which they already mined one of the previous *spacing-1* blocks.

---

[20] This risk could be reduced by introducing a miner "heartbeat", in which each miner indicates their presence on the network by regularly broadcasting a transaction that moves no funds.

[21] The effect of different *mining diversity* values can be calculated if we assume that each miner has an independent probability *f* of technical failure and *c* of malicious collusion. The probability Pr(F) of a mining freeze is given by the cumulative binomial distribution Pr(Bin(*miners*, 1-*f*) ≤ (*spacing*-1)). Similarly, the probability Pr(C) of malicious miners secretly mining an alternative chain is Pr(Bin(*miners*, 1-*c*) ≤ (*miners*-*spacing*)). For example, if *miners*=20, *f*=2.5% and *c*=25% then setting *diversity*=0.75 keeps both Pr(F) and Pr(C) below 0.001%.

As well as preventing abuse, the diversity threshold helps in a case where the network splits temporarily into disconnected islands, perhaps due to a communications failure. This mishap will lead to a fork in the chain, as each island is unable to see the other's transactions and blocks. Once the network is reunited, the fork with the longer chain will be adopted as the global consensus. The diversity threshold ensures that the longer blockchain will belong to the island containing the majority of permitted miners, since the other island's chain will quickly freeze up.

If mining is restricted to certain entities, one might question the advantage of a private blockchain over a centralized database which accepts incoming transactions, resolves disputes, and answers queries regarding the database's state. The answer is threefold:

- Each participant retains full control over its assets via their private key. Even miners cannot create transactions that spend another party's funds.
- Control of the database is distributed across many entities, so that no individual or small group can unilaterally decide which transactions are valid or will be confirmed.
- Superior robustness, since the disappearance or malfunctioning of one server will not affect the continued processing of transactions by the network as a whole.

Where does permissions-based mining leave proof of work? Recall that in bitcoin, proof of work ensures mining diversity by making it computationally difficult (and therefore costly) to create a block. By contrast, private blockchains use a much simpler scheme for enforcing mining diversity, so this "work" can become little more than a formality. In practice, the first version of MultiChain still uses a bitcoin-style proof of work to regulate and randomize each node's rate of block production, but this is not the basis of the blockchain's security[22].

In a MultiChain blockchain, transaction fees and block rewards are zero by default. If the cost of mining a block is negligible, miners need no compensation for providing this service beyond their general stake in the blockchain's smooth functioning. Alternatively miners might charge network participants a fixed annual service fee, paid by traditional off-blockchain means. If a blockchain's sole purpose is to enable transactions in tokenized assets, its "native" currency might safely be ignored as an evolutionary artifact. However, if transaction scarcity is desired, MultiChain can also be configured to use a native currency for block rewards, minimum transaction fees and output quantities. In this case participants would need to purchase units of the native currency from miners, perhaps in exchange for a tokenized asset.

Multiple configurable blockchains

Rather than supporting a single blockchain like Bitcoin Core, MultiChain is easy to configure and can work with different blockchains at the same time. The immediate benefit for institutional users is enabling private blockchains to be configured and deployed by system administrators rather than specialized developers. An analogy is the way in which relational database management systems such as Oracle or SQL Server allow databases to be created and used with a few SQL commands. A further benefit of supporting multiple blockchains is the ability for a server to create connections between the activity in different chains. For example, an institution may want the arrival of funds on one blockchain to trigger a corresponding transfer of funds on another.

---

[22] Some blockchains use proof of stake, in which mining rights are granted to participants in proportion to their holding of the network's native currency. A future version of MultiChain may also offer this option.

MultiChain allows the user to set all of blockchain's parameters in a configuration file, including:

- The chain's protocol, i.e. private blockchain or pure bitcoin-like.
- Target time for blocks, e.g. 1 minute.
- Active permission types, e.g. anyone can connect, only some can send/receive.
- Mining diversity (private blockchains only).
- Level of consensus required for creating/removing administrators and miners, and the duration of the setup phase in which this is not enforced (private blockchains only).
- Mining rewards, e.g. 50 native currency units per block, halving every 210,000 blocks.
- IP ports for peer-to-peer connections and the JSON-RPC API, e.g. 8571, 8570.
- Permitted transaction types, e.g. pay-to-address, pay-to-multisig, pay-to-script-hash.
- Maximum block size, e.g. 1 megabyte.
- Maximum metadata per transaction (`OP_RETURN`), e.g. 4096 bytes.

Multiple blockchains can be active on a single server, each with its own name and configuration file. To create a new blockchain, two simple user steps are required. First, the user chooses a name for the chain, upon which MultiChain creates a configuration file containing the default settings. This file can be modified by the user, although the defaults will be suitable for common use cases. Second, the user launches the blockchain, upon which the genesis block is mined by MultiChain, granting its creator all user privileges. At this point, MultiChain also embeds details of the genesis block along with a hash of all the blockchain's parameters in the configuration file, in order to prevent subsequent accidental changes.

When first launched, a blockchain runs off a single node only. To add a new node, MultiChain is run from another computer with three parameters: (a) the destination blockchain name, (b) its IP port number, and (c) the IP address of an existing node. For user convenience this information is combined into a "node address" in a familiar form, e.g. `chain1@12.34.56.78:8571`[23]. At first a new node will not be permitted to connect, since the network is private and the node has not yet been granted connection privileges. MultiChain will display a message containing the new node's self-generated public address, which must be sent to an administrator. The administrator grants connection privileges to this address via a simple command that creates the appropriate transaction. The new node can then reconnect successfully and automatically downloads the configuration file defining the blockchain's characteristics. Any future connections to the same blockchain only require the chain name to be specified, with the handshaking process between nodes ensuring that they use identical parameters.

An obvious future enhancement is to allow some parameters to be changed once a blockchain is running, via special transactions issued by trusted administrators. For example as usage of a network grows, the maximum block size could be increased to accommodate the expected volume of transactions. Any such changes must take into account the computational capacity of each of the nodes on the network.

Multicurrency blockchains

Recall that tokenization protocols such as CoinSpark and Counterparty enable third party assets to be issued and transacted over the bitcoin blockchain, in parallel to bitcoin's native currency. These techniques can equally be used on private blockchains created by MultiChain, without further

[23] The node address is displayed whenever MultiChain creates or connects to a blockchain and can easily be sent over email or dictated over a phone call.

modification. However, in a blockchain running a private protocol, we can improve on these schemes by integrating support for third party assets directly into the chain's rules.

In bitcoin, every transaction encodes the quantity of bitcoin contained within each of its outputs. If a transaction has more total bitcoin encoded in its outputs than the total coming into its inputs, it will be considered invalid by the network and not propagated or confirmed on the blockchain. This validation is possible because every network node tracks the quantity of bitcoin in unspent transaction outputs. As a result, the presence of a transaction in the network or blockchain is sufficient to give users confidence regarding the accuracy of the bitcoin quantities encoded within. In turn, this enables lightweight ("simple payment verification") wallets to transact safely with the network, without storing the entire blockchain on the user's computer.

The problem with asset tokenization over bitcoin is that the metadata which encodes the presence of non-native assets is not subject to this same network-level verification. Let's imagine that ABC bank has issued tokens representing dollars. A rogue user can create a transaction whose metadata indicates that it contains 100 ABC dollars in an output, even if no ABC dollars were present in that transaction's inputs. This transaction will be accepted as valid by the bitcoin network and confirmed on the blockchain, because (a) bitcoin nodes cannot read this metadata, and (b) bitcoin nodes do not track ABC dollars.

Tokenized assets are therefore second-class citizens on the bitcoin blockchain, in comparison with the chain's native currency. The presence or otherwise of a tokenized asset can only be calculated by examining the full history of all transactions affecting that token since the transaction which created it. This can be calculated efficiently in a "forwards" manner, examining every new transaction as it comes in. Nonetheless it still requires a full network node, undermining the suitability of tokenization protocols for use with lightweight wallets[24].

MultiChain solves the problem by encoding the identifiers and quantities of all assets into each transaction output, using an extension provided by bitcoin's scripting language[25]. The rule for validating transactions is extended to verify that the total quantities of all assets in a transaction's outputs are exactly matched by the total in its inputs. This equality requirement is stricter than the constraint for the native currency, in which the output may be less than the input, with the difference collected as a fee by miners. Of course, the blockchain must also allow the creation of new assets in a "genesis" transaction with special metadata. MultiChain automatically assigns identifiers for new assets based on the position of this genesis transaction in the blockchain, alongside a user-defined unique textual identifier.

MultiChain's permissions system can be used to restrict the right to create assets. In addition, future versions could introduce per-asset permissions, in which each type of asset has its own set of administrators and permitted senders and recipients. For the sake of simplicity this feature is not included in the first version of MultiChain. However its inclusion would only require a simple extension to the rules already implemented.

---

[24] CoinSpark mitigates this problem via issuer-provided asset tracking servers that can be queried by wallets. Unlike most tokenization protocols, one CoinSpark asset can be tracked while ignoring all others. While this solution is practical, it still creates a burden on issuers and a delay in updating end user asset balances.

[25] MultiChain uses bitcoin's `OP_DROP` opcode to encode metadata with transaction outputs.

Bitcoin to private blockchains and back

As Internet usage exploded in the 1990s, it exposed millions of people to new paradigms for communication and collaboration. Enterprises wished to deploy these innovations internally, however many did not believe that the Internet would offer sufficient privacy, reliability or capacity to serve as their communications backbone. As a result many enterprises built miniature internal versions of the Internet called Intranets, which deployed the same technologies over infrastructure that was entirely under their control. A decade or two later, the Internet had consolidated its position as a robust high capacity global network for transmitting information. This enabled many enterprises to make use of virtual private networks (VPNs), which use the Internet as a backbone but encrypt the organization's traffic over these public pipes. VPNs allow enterprises to enjoy the Internet's economy of scale, while ensuring that their data is not visible to outside observers.

One can imagine a comparable process playing out between the bitcoin blockchain and private blockchains. From the enterprise's perspective, the bitcoin network is currently frontier territory – wild and uncontrolled, with limited capacity and unpredictable long-term transaction costs. Worst of all, bitcoin mining is controlled by largely unknown parties, many of whom are ideologically opposed to corporations or are located in countries with weak legal systems. Therefore private blockchains will likely be a more attractive solution for financial institutions wishing to deploy this technology during the next decade. Twenty years from now, if bitcoin or another blockchain is processing billions of transactions monthly at very low cost, with mining controlled by large identifiable corporations, bitcoin may start looking like an attractive platform for institutional financial transactions. As with VPNs, a thin encryption layer could be used to ensure that institutional activity is hidden from the majority of network participants.

Many aspects of MultiChain's design are aimed at enabling smooth transitions between private blockchains and the bitcoin blockchain in either direction:

- MultiChain is based on a fork of Bitcoin Core, the official client for the bitcoin network. Code changes are localized, enabling future bitcoin enhancements to be merged in.
- It uses bitcoin's protocol, transaction and blockchain architecture, with changes only to the handshaking process when two nodes initially connect. All other features are implemented using metadata and modifications to the validation rules for transactions and blocks.
- Its interface (command-line and API) is fully compatible with that of Bitcoin Core, with all additional functionality provided by new commands.
- It can act as a node on the regular bitcoin network (or other bitcoin-like networks), via a simple protocol setting in the per-blockchain configuration file.
- Its multicurrency and messaging features (see later) work very similarly to the CoinSpark protocol for enhancing bitcoin transactions.

This last point ensures that an application which uses asset tokenization and messaging can be moved between bitcoin and private blockchains with minimal changes to its code.

MultiChain roadmap

Below is a list of some additional features that may be added to MultiChain in future.

Blockchain messaging

A private blockchain can provide authenticated and notarized messaging, using a similar approach to the CoinSpark protocol for bitcoin. There are two main uses for messaging. First, it can be used to add important context for an on-blockchain financial transaction, such as a contract, receipt or invoice. Alternatively, it can be used for pure notarized communication with no related movement of funds. In either case, information on the blockchain enables both the sender and recipient to prove the timing and content of the correspondence that took place.

Each message has an associated hash, which is a long number that uniquely fingerprints its content. Given any input data and a particular hashing algorithm, it is easy to calculate the corresponding hash. However, for any secure hashing scheme, it is practically impossible to generate a piece of data to match a given hash. Secure hashing algorithms are therefore called one-way functions, since they can only be calculated in one direction.

A message is sent from the originator to the recipient as follows:

1. The originating MultiChain node sends a message transaction to the recipient with metadata containing its IP address and a hash of the message's content[26].
2. The receiving node receives the transaction and decodes this metadata.
3. The receiving node contacts the originating node via its IP address to retrieve the message, signing the request in order to prove its identity as the intended recipient. This communication take place using the blockchain's existing peer-to-peer protocol.
4. The receiving node verifies the message's validity by checking its hash against the hash embedded in the original transaction.
5. If the message is valid, the receiving node completes the loop by sending back a second transaction to the sender containing the same message hash.

Once the first transaction is confirmed on the blockchain, the recipient can prove: (a) who sent the message, since the sender reveals their address when signing the transaction, (b) the time the message was sent, since the transaction is embedded in a timestamped block, and (c) the message's content, since the hash is part of the transaction that was signed. However none of this is sufficient for the sender to prove that a particular message was received by the recipient. Indeed, malicious senders could even embed a hash of one message while sending a completely different message in step 3 above. Therefore we require a second transaction in which the recipient sends back a receipt containing the same hash. Once this transaction is confirmed, both parties can prove all the details of the correspondence that took place.

As with public messages in CoinSpark, such a system could also be used to broadcast information openly to all network participants, with the message hash on the blockchain serving as proof of the message's content and publication time. In this case we simply skip the restriction on who may retrieve the message from the originating node, and do not require a second transaction to confirm the message's receipt.

[26] Each message also contains a unique random "salt" which affects the hash but is not displayed to the user. This prevents message snooping via dictionary attacks: http://en.wikipedia.org/wiki/Salt_(cryptography)

Decentralized exchange

The per-output transactional model used in bitcoin enables the creation of transactions in which two (or more) parties exchange some assets safely. The blockchain treats this transaction as atomic, meaning that it either succeeds or fails as a whole, so there is no risk of one party losing their asset without receiving the corresponding asset from the other side. This is the equivalent of delivery-versus-payment in the world of traditional financial settlement.

As an example, let's consider a simple two-way exchange between $15 belonging to Alice and £10 belonging to Bob. The exchange is performed in a transaction with two inputs and two outputs. The first input comes from Alice and contains her $15, while the second input comes from Bob and contains his £10. The first output goes to Alice and contains £10, while the second output goes to Bob and contains $15. As discussed earlier, MultiChain enables these asset values to be encoded directly into transaction outputs, so the match between input and output quantities is verified by every node in the network. Assuming the transaction is valid and properly signed by both Alice and Bob, it will be propagated and accepted onto the blockchain.

Unfortunately the process of constructing such a peer-to-peer exchange transaction is rather clumsy, consisting of the following steps:

1. Alice and Bob discover each other's mutual willingness to perform the exchange through some off-blockchain process, and agree on the transaction outputs they will use.
2. Alice's node constructs the full transaction and signs it.
3. Alice transmits the partially signed transaction to Bob via another off-blockchain process, since this incomplete transaction will not be accepted by the network.
4. Bob receives the transaction and verifies that it accords with their agreement. If so, his node signs the transaction as well. At this point the transaction is valid.
5. Bob's node transmits the fully signed transaction to the network which verifies it and confirms it on the blockchain.

This process contains two steps which take place outside of the network – first, for Alice and Bob to find each other and second, to send the partially signed transaction from Alice to Bob[27].

A future version of MultiChain will streamline this process by enabling partial transactions to be directly propagated across the network. Such partial transactions represent an offer for exchange, which any party can accept by completing the transaction and transmitting it for inclusion in the blockchain. Fortunately this doesn't require any significant changes to bitcoin's transactional model and signing process, since bitcoin already has a method for users to create and sign a partial transaction, allowing the content of other inputs and outputs to be changed[28].

We can see how this works in practice by following the example above. First, Alice's MultiChain node constructs a transaction in which the first input comes from Alice with $15, and the first output goes to Alice containing £10. This transaction also contains a second input and output, which are left as empty placeholders. Alice's node signs the first input and output of this partial transaction in order to render this part valid. This partial transaction will not be accepted onto the blockchain, due

[27] As a partial solution, transmission of the partially signed transaction could be performed using the secure messaging protocol outlined above. But this still does not help Alice and Bob to meet.

[28] The technical term for this type of signature is `SIGHASH_SINGLE`.

to the mismatch between the asset quantities in its inputs and outputs. However MultiChain will still allow it to be propagated across the network, and held in the temporary "memory pool" storage area of each node. When Bob's MultiChain node receives this partial transaction, it presents it to him as a possible exchange, and Bob accepts. Consequently Bob's node completes the transaction by setting the second input to Bob's £10, the second output to $15 destined for Bob, and signing the entire transaction[29]. Now the transaction is valid, since it contains the same total asset quantities in both inputs and outputs. Bob's node retransmits it to the network and it can be confirmed.

In this system Alice needs a way to cancel her offer after it has been broadcast. She does this by creating a new transaction which spends the output referenced by the input of her offer, sending it back to herself. This renders the offer invalid, since it spends a transaction output that is no longer available, so it will be discarded by all network nodes. It should be noted that this type of decentralized exchange is not suitable for high frequency trading (HFT) applications, due to the propagation delays inherent in a peer-to-peer network. In addition care must be taken to ensure that one participant cannot flood the network with unattractive offers. In a private blockchain this can be achieved by restricting the right to make offers to pre-specified user addresses, and introducing a limit on the number of outstanding offers created by each address.

Database synchronization

Blockchains are an outstanding technology for ensuring that all participants in a decentralized network share an identical view of the world. However they are poorly optimized for answering queries regarding past activity on that network, since they represent that activity as a raw log of confirmed transactions. This log is stored in chronological order, grouped by block number, without any additional indexes. There are many useful reports we may want to generate, such as listing all activity for a particular address or asset, which can only be answered by scanning the entire blockchain to search for matching transactions.

To solve this problem, a future version of MultiChain will include a bridge between its blockchains and regular relational databases such as Oracle, SQL Server or MySQL[30]. This bridge will treat each blockchain as the master of a replication process in which the relational database is the slave. The database will also reflect new unconfirmed transactions, by treating the node's memory pool as a special "pending" block. By using ordinary database indexing techniques, the transactions in this database can be analyzed efficiently using regular SQL queries.

Deployment scenarios

As a general purpose platform for private blockchains, MultiChain can be deployed for a wide range of use cases. In this section we provide three example deployment scenarios, and suggest the appropriate permissions and mining diversity to use for each case.

Centralized currency settlement

Let's begin with a simple case, in which a custodian for regular currencies uses a private blockchain for itself and its clients, instead of a regular database combined with customer-facing APIs. This use

[29] Bob can also use an input containing more than £10, including other assets. In this case he includes any remaining funds in the second output which is sent back to him, and the transaction remains valid.

[30] Blockchain explorers such as blockchain.info are based on the same principle.

of a blockchain does not change the custodian's business model but rather serves to reduce IT costs and settlement delays.

In this scenario, the custodian acts as the sole administrator, miner and issuer for the blockchain, but distributes these functions across several MultiChain nodes for robustness and redundancy. The mining diversity parameter is set to zero, since all mining is controlled by a single trusted entity, and there is no problem if just one of that entity's nodes mines all the blocks, perhaps due to other nodes failing.

After creating the network, the custodian begins by granting its clients the permission to connect and transact on the blockchain. It then creates tokenized assets for the different currencies to be transacted. These tokens are sent to clients in exchange for a corresponding deposit of cash in the custodian's bank account, and represent the right to redeem that cash from the custodian at any time. The clients can then send cash directly to each other using transactions which move the corresponding tokens. Changes in ownership represented by transactions are finalized ("settled") once those transactions are confirmed on the blockchain by one of the custodian's mining nodes.

As well as straightforward payments, this blockchain could also be used as a mechanism for transparent peer-to-peer currency exchange, with the custodian being responsible for settling the exchange transactions. As discussed earlier, a client could create a partially signed transaction representing an offer of exchange, perhaps between dollars and euros, and distribute it across the network. Any other client could then accept the exchange by providing the missing input and output and transmitting the completed transaction.

Despite this being a centralized system, there are several advantages to using a blockchain over a regular database. First, the blockchain provides a single unified view of the state of play, so clients have no need to maintain separate records. Since there is no possibility of disagreement over the nature of a transaction, no reconciliation is required and trading breaks cannot occur. In addition, settlement times are drastically reduced to the time that it takes to mine a block. A further advantage is that the system is highly fault-tolerant, with dense peer-to-peer connectivity between nodes and many custodian miners providing redundancy.

Bond issuance and peer-to-peer trading

Let's consider a second case, in which several financial institutions collaborate to build a blockchain-based network for trading in newly-issued corporate bonds. Whereas previously these bonds might be issued via an underwriter and traded on an OTC (over-the-counter) basis, a blockchain enables both of these activities to take place more quickly and transparently.

In this scenario, there is no centralized control over administration or mining. Instead, each institution participating in the network has mining and transaction privileges, with a small number of "senior" participants holding the administrative rights to assign privileges to others. The mining diversity is set to a high value such as 0.9, meaning that the consensus could only be undermined by at least 90% of the permitted miners collaborating secretly and maliciously. In addition, legal provisions are put in place so that such an event would represent a severe breach of contract, with the injured parties seeking remedy in court.

The right to create assets is granted only to companies issuing bonds on the blockchain, for a short window around the time of bond issue. The origination of a bond is represented by the creation of a new tokenized asset by the debt-raising company. Tokens of this asset are sent to lenders in exchange for cash transferred to the company. If we prefer this cash transfer to also take place on

the blockchain, an additional currency asset could be issued by a trusted party and used for this purpose. As in the previous scenario, participants could buy and sell these currency tokens using deposits and withdrawals in the issuer's bank account.

The rights pertaining to bond holders are legally defined in terms of the state of the blockchain at particular points in time. For example, interest payments are made based on ownership of the asset in the first block whose timestamp is after their due date. At maturity, bonds are redeemed by their tokens being sent to the issuer in exchange for a cash payment to the bondholder.

Each exchange involving bonds can take place atomically, using a single blockchain transaction that represents the exchange. However, if mining is conducted collaboratively by competing institutions, partially signed transactions cannot be used to implement a decentralized exchange over the blockchain network. Let's imagine that an attractive offer of exchange is transmitted as a partial transaction, leading several participants to create complete transactions which accept that offer. In this case, the miner of the next block has the power to choose which of these competing transactions is confirmed. If the miner has a financial stake in the outcome of that contest, perhaps because they created one of those transactions, a clear conflict of interest arises. Indeed, because of the constraint on mining diversity, participants might avoid mining altogether until a transaction appears which they want to see confirmed. As a result of these risks, the matching of parties in exchange transactions needs to take place in an external process, with the blockchain serving only to rapidly settle those exchanges.

Consumer-facing rewards scheme

Let's broaden the application of blockchains to include a consumer-facing element, in which several US restaurant chains band together to create a rewards scheme. This scheme enables the rewards collected in one restaurant to be used in another, without giving centralized control of the database to any single company or external contractor.

This use case introduces a distinction between the core and periphery of a blockchain network. The core consists of regular nodes of MultiChain, which store the entire blockchain and verify all transactions and blocks as they come in. At the periphery are lightweight wallets, which can also transact over the network but do not store the blockchain or verify transactions and blocks.

Each company participating in the rewards scheme runs a full MultiChain node which has administration, mining and asset creation privileges. The mining diversity parameter is set to approximately 0.75, to allow some nodes to fail without freezing the blockchain. Beyond this, the blockchain allows unrestricted access for connecting, sending and receiving transactions.

Consumers in the scheme use lightweight wallets, running as mobile apps, which connect to several full nodes in order to receive and send transactions. The inline encoding of asset quantities within transaction outputs enables these lightweight wallets to transact safely over the network without needing to separately track the movements of assets. Since the blockchain has no restriction on connecting, sending and receiving privileges, anyone can begin using the system by installing a mobile wallet which generates its own private key and address.

The assets themselves are vouchers, issued in each company's name and denominated in US dollars. These vouchers are given to customers as rewards for purchases in the participating restaurants. They can then be redeemed by customers at sticker price for purchases at any of the restaurants participating in the scheme. Finally, each company is able to redeem the vouchers issued by another company in exchange for cash at (say) 30% of sticker price.

Ensuring privacy

In any blockchain, all transactions are publicly viewable to all participants, and this creates a fundamental problem in terms of privacy. First, blockchains enable each participant to gain a global picture of the aggregate volume of assets being held and traded. Depending on the use case, this transparency may or may not be desirable. However a more serious problem is that participants learn the public addresses of other participants when transacting with them, enabling them to infer their counterparty's full balance and trading activity in both the past and future.

The simplest way to resolve this problem is for each participant to transact under many different addresses. When sending or receiving transactions, they can use a different address depending on the identity of the counterparty. This prevents either party from gaining a full picture of their counterparty's activities, since they do not know which other addresses the counterparty is using. Participants can move assets between their addresses as and when is required, taking care to ensure that those transactions are indistinguishable from payments to other entities. Alternatively, a trusted central party could provide a "coin mixing" service, allowing assets to deposited and subsequently withdrawn using different addresses, while ensuring there is no visible connection between the deposit and withdrawal transactions.

A more fundamental approach to privacy is promised by cryptographic techniques such as homomorphic encryption and zero-knowledge proofs. In a general sense, these enable specific computations to be performed, with their accuracy publicly proven, without revealing the inputs and outputs of those computations. In a blockchain the techniques can be applied to hide a transaction's asset quantities from all but the sender and recipient of that transaction, while still enabling all network participants to verify that the transaction is valid[31]. If blockchain participants cannot see the quantities in each other's transactions, it becomes trivial to obscure genuine activity behind a smokescreen of transactions which move negligible amounts.

Acknowledgements

Thanks to Toby Coppel, Rachel Lee, Simon Liu, Michael Rozantsev, Lior Yaffe and Maya Zehavi for helpful comments and feedback.

Revisions

June 2015: First revision.
July 2015: Paragraph about cryptographic schemes for obscuring asset quantities.

[31] See: https://people.xiph.org/~greg/confidential_values.txt and http://voxelsoft.com/dev/cct.pdf for discussion.