BitTribe: A Peer-to-Peer Monetary System

-In Memory of Satoshi Nakamoto

BitTribe Native Residents
Followers of Satoshi Nakamoto's Original Vision
residents@bittribe.io
www.bittribe.io

October 31, 2018
（Version 0.1）

Abstract. The Bitcoin System by Satoshi Nakamoto is enthusiastically celebrated for two reasons: 1) it has proven for the first time in human history that a peer-to-peer public bookkeeping ledger is not only technically feasible but also practically robust; and 2) it has pioneered the exploration on money issuance in a peer-to-peer system. While the first achievement is almost indisputable despite a few minor criticisms, the second experiment is less successful and believed to be still an unfinished cause. Bitcoin, a crypto asset, acclaimed as gold in the virtual world and deemed to have unparallel collection value, fails to feature stable exchange ratios relative to the goods and services within any specified domain in either the virtual or the real world, which is why the Bitcoin System may not suit to serve as the foundation of a meaningful monetary system on a global scale. We, however, believe that the idea behind the Bitcoin System naturally leads to the vision of building a peer-to-peer monetary system, called BitTribe, of which the core consists of both a fully decentralized public ledger and a stable hence more useful money. This paper explores the logical elements necessary to achieve such a vision and proposes an open project soliciting efforts in the global community to work on it.

1. Introduction

The Bitcoin System is a splashing success in that it has provided a peer-to-peer public bookkeeping ledger and the resulting crypto asset Bitcoin is considered digital gold by many people. However, a ledger by itself alone is not enough to serve as the foundation of a monetary system, as the high volatility of Bitcoin making it almost impossible to act as an everyday payment means for goods and services in the real world. It takes in addition a money that is stable in value to accomplish the mission.

While the whitepaper "Bitcoin: A Peer-to-Peer Electronic Cash System" by Satoshi Nakamoto focuses on the mathematics and computation logics of a peer-to-peer ledger, we hereby try to come up with a sequel to the seminal paper by elaborating possible modifications and extensions of the Bitcoin System such that the peer-to-peer mechanism may be applied more practically in the monetary domain. In particular, we want to explore the possibility for issuing a stable money in a peer-to-peer system "without going through a financial institution" (words by Satoshi Nakamoto). In short, we try to build a peer-to-peer monetary system, i.e., a BitTribe.

Our pursuit for BitTribe is inspired by Satoshi Nakamoto's theory and practice, and more importantly, by his vision as we believe. Before we discuss how to build a BitTribe, let us review the pros and cons of the Bitcoin System.

The public ledger in the Bitcoin System, notably based on the proof-of-work consensus, is regarded widely as a masterpiece. But there are still criticisms about it and the concerns below are mentioned frequently.

First, the energy consumption of the system is astonishingly high. This may be a valid point but does not necessarily bear a significant weight to some people, as a peer-to-peer public ledger has its social benefits that may by far outweigh the energy costs.

Second, the computing power is now concentrated in a few hands controlling the top mining pools. The dispersion of peers, which is crucial to the safety and robustness of the ledger, may be compromised. The concern may be valid even when the probability of a successful fraud is minimal, in that the cost to recover the system from a disruption by a failed fraud back to its normal working order may be too high to bear. Moreover, the concentration of computing power leads to the concentration of mining rewards, a sociological inequality that is a potential threat to the stability of the system.

Another major shortcoming of the Bitcoin System is due to the fact it does not feature a stable money. Bitcoins, i.e., tokens generated within the system as rewards to the ledger maintainers or miners, are not a good money despite the misleading name with "coin" in it. The pre-determined formula generates a constant number of Bitcoins per unit time and does not accommodate changes of sales or transaction volume in any specified economy. As a result, the value of Bitcoin does not stay stable relative to any economy.

By the above discussion, we conclude that for a BitTribe,

a) The concentration of computing power as in the Bitcoin System may not be tolerable as a monetary system, and

b) The lack a stable money as in the Bitcoin System must be redressed.

We hereby propose an open project to pursue a proper way to build a BitTribe. More specifically, we have the following goals:

a) Discuss the theoretical framework for a BitTribe, i.e., a peer-to-peer monetary system which consists of a public ledger and a stable money;

b) Discuss possible implementation schemes of the above theoretical framework.

2. The Essential Elements of a BitTribe

A BitTribe is a set of virtual residents, with a peer-to-peer protocol governing their communications, and featuring the following properties:

a) ***A Decentralized Crypto ID*** It supports autonomous crypto ID verification without a central authority. The pair of private key and public key used in the Bitcoin System is decentralized while all verifications require user's direct interaction using the private key. This is not practical because a resident may not be online all time and be able to answer the verification inquiry without advance notice in a timely fashion.

b) ***A Public Ledger*** It supports a bookkeeping ledger that is maintained by all or some of the residents bound by a peep-to-peer consensus rule. It is required to be technically safe and robust; and it is desirable to have the mining rewards distributed as diversely as possible among the participating maintainers or miners.

c) ***Support of Token Issuance*** Tokens can be issued by using smart contracts. Note that the smart contracts are to support token issuance and token transactions only, but may not necessarily need to support more complicated business applications.

A token is usually used to represent a set of certain rights and interests. Generally speaking, rights and interests may be generated endogenously within the system（such as a Bitcoin） or exogenously imported from outside of the system (such as a certificate of stocks or real estate properties) .

d) ***Support of Money Issuance*** Money may be treated as a special token that has a stable value relative to certain set of reference assets. As this is a peer-to-peer world, the money issuance is done by a smart contract. We refer to this particular smart contact as the ***Money Issuance Protocol***, similar to the money issuance policy deployed by a ***central bank*** in the real world. More specifics about the stabilizing logics and mechanisms will be discussed in the following sections.

e) ***Capability to Communicate with External World*** A BitTribe should be able to communicate with the external world, i.e., another BitTribe or the real world. Specially, we need it to support the following:

- Allow the external world read data from the BitTribe and the data can be decrypted with proper authorization;
- External variables can be imported into the BitTribe via Oracles; and
- Cross-chain communications are supported. （To be discussed more.）

With the above features, applications upon a BitTribe can be developed. Note that a particular application may be realized with one or more smart contracts, or some smart contracts interacting with an external centralized system through its API. For example, a depositing and lending facility can be set up (with deposit and lending interest rates) with a smart contract, which may be compared to a ***commercial bank*** in the real world.

A group of applications, if inherently related, can be viewed as a family. The residents running and using the services of a particular family may also set up additional rules governing themselves, which form a clan within the BitTribe.

On the other end, we expect that there will be more than one BitTribe in the world. At certain points of time, some protocols are needed to coordinate the following affairs:

a) Cross-tribe identity authentication;

b) Cross-tribe money payments;

c) Cross-tribe smart contract calls.

The set of above protocols, together with the infrastructure to implement them (if necessary), should be and can be peer-to-peer.

In the rest of this paper, we are to elaborate on the above elements one by one. At certain points we may also propose specific implementation suggestions. However, we want to warn the reader that any suggestions hereby may be tentative, only to serve as an example illustrating the views and soliciting further discussion.

3. Decentralized Crypto ID

The current private/public key pair used in the Bitcoin System is decentralized. However, verification of user's identity requires signature using user's private key every time. This is not very practical because user cannot be always online to provide signature without notice and cannot afford precious time on frequent identity verifications. Thus, in many practical use cases, a universal ID crossing many platforms, cryptocurrencies, blockchains, exchanges, wallets and banking systems is indeed needed for autonomous verification of crypto identity. We define a ***universal decentralized crypto ID, DID***, based on ***decentralized PKI*** and ***Verifiable Random Function (VRF)*** standards as the following,

a) User uses his/her ***personal online identifier, PID***, such as email address or mobile phone number as the input to generate a unique hash to protect the privacy of original PID,

$$\text{DID} = \boldsymbol{hash_{PID}} = \text{hash(PID)}$$

b) User's hash_{PID} and user's ***private key SK*** generate a ***VRF hash*** and its ***VRF proof***,

   Beta_{DID}** = **VRF_hash(SK, DID)

   Pi_{DID}** = **VRF_prove(SK, DID)

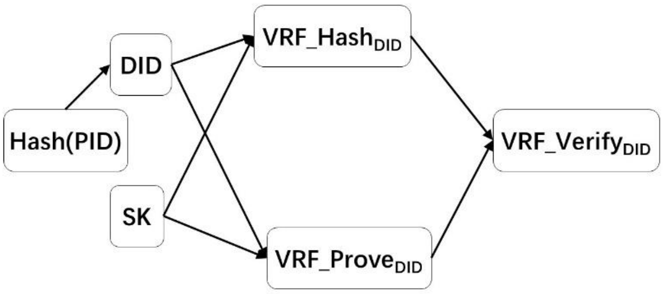
c) User sends a transaction on blockchain to store his/her beta_{DID} and pi_{DID} so DID can be verified with his/her ***public key PK***,

   VRF_verify(PK, DID, beta_{DID}, pi_{DID})

d) Other user can verify a user's DID by sending a transaction on blockchain to this user's ***public address PA_{USER}*** described below to validate the DID along with its public key PK

e) User could change his/her public DID by repeating the steps 1 to 3 using a different PID, e.g. another email address or a new phone number

This universal DID can be used in the entire crypto world.

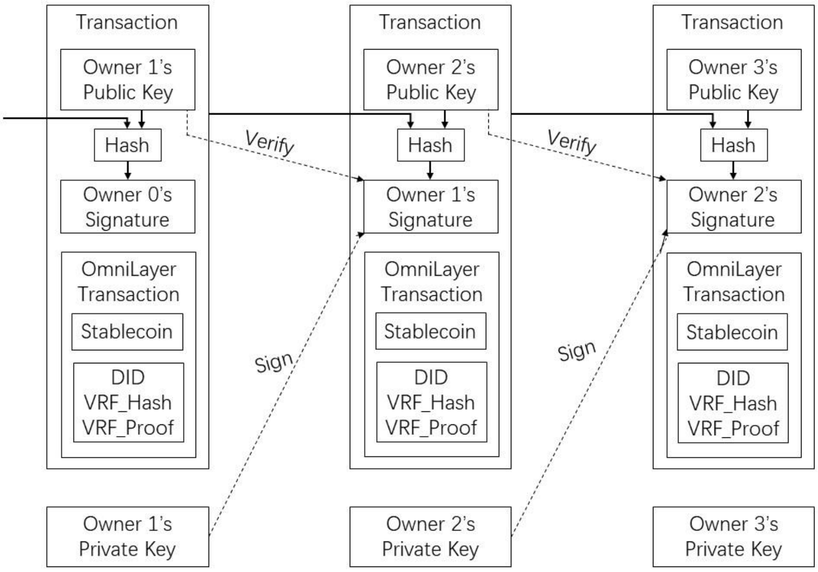


Another practical use case is to use a user's public address on blockchain, PA_{USER}, to replace $\text{alpha}_{\text{PID}}$. In such case, the user's public key PK can be verified through its public address PA_{USER}. Of course, this will weaken the VRF security because the public address PA_{USER} is known. But it's an acceptable risk because it will not be used for identify verification. These two identity verifications could be used together where public address PA_{USER} is used to verify user's public key while DID is used to verify user's identity. Repeating the above process to generate the VRF hash and VRF proof of public address PA_{USER} by replacing $\text{alpha}_{\text{PID}}$ with PA_{USER}.

The distributed VRF hash, its VRF proof and PA_{USER} based on decentralized PKI and VRF standards form a foundation of verifiable user identity in the crypto world without direct interaction with user and at the same time protect the privacy of user's private identity in the real world. A balance between privacy and verification.

4. Transaction

The transaction at the blockchain level is identical to that in Bitcoin. However, we have added a sub transaction at ***OmniLayer*** level to support the implementation of money and decentralized crypto ID, DID, without alerting the blockchain transaction at the blockchain level. The ***OmniLayer transaction*** is executed within the ***OP_RETURN*** script code.



The ***OmniLayer Specification*** offers much more functionalities beyond the implementation of money and DID. Here is a summary of its functionalities:

a) Transferring Coins (use for money)

b) ***Distributed Exchange*** (***decentralized exchange***)

c) ***Smart Property*** (types of crypto properties other than coin)

d) Smart Property Administration

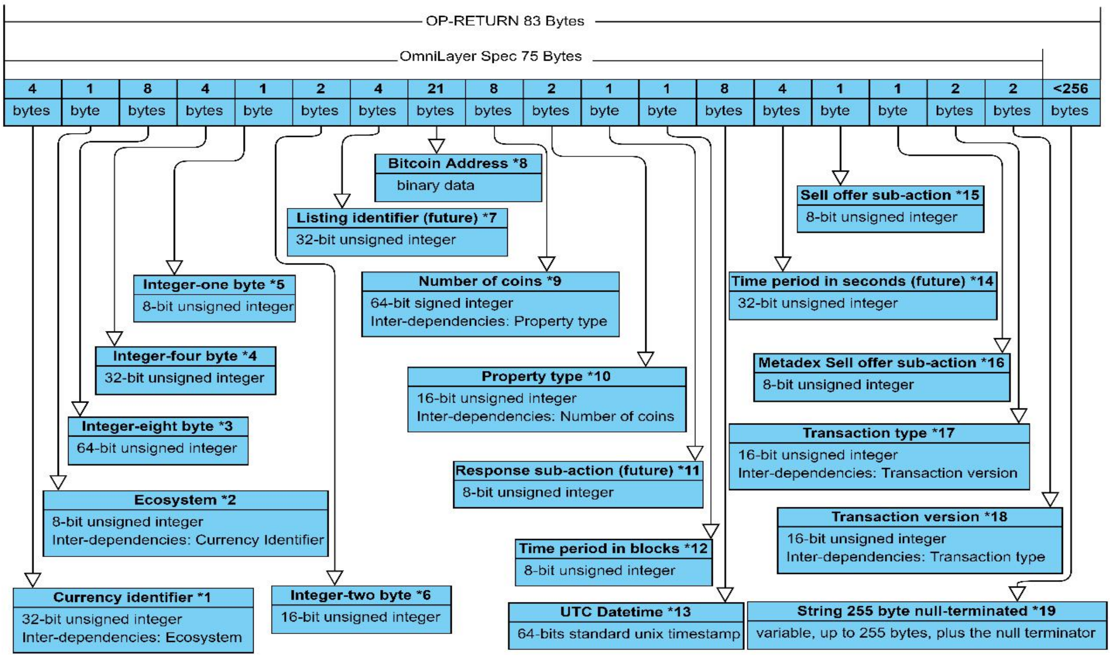
e) Future Transactions (undefined)

f) ***Distributed E-Commerce*** (decentralized exchange of Smart Property using stablecoin)

g) ***Escrow-Backed User Currencies*** (experimental proposed feature)

h) Extendable to any type of events such as p2p messaging, group messaging, publishing, subscription, other types of p2p communication, etc.

i) Our extension to support ***DID transaction***



It offers the similar functionality of Smart Contract with any predefined type of transactions without using a ***Virtual Machine (VM)***. Of course, it does not support arbitrary or dynamically defined transactions that a generic programming language can define. Because it does not use VM so it does not have the attacking space exposed by VM and generic programming languages. For any given type of transaction and contract, if it can be done with OmniLayer sub transaction, it is significantly safer than VM based Smart Contract.

5. Peer-to-Peer Ledger and Mining

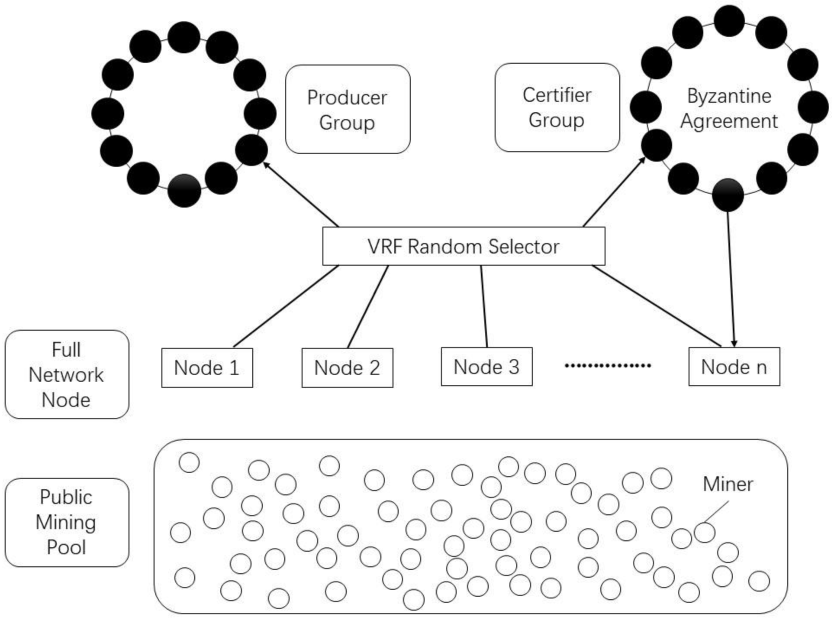
The ledger maintenance in a BitTribe may be constructed following the general idea of Satoshi Nakamoto, i.e., motivating miners to compete for the leading role in keeping the book. The reward token is called an ore. A BitTribe may be named after the ore, say a Diamond BitTribe, a Jade BitTribe, etc.

When designing the mechanism, two issues may be kept in mind to improve upon the Bitcoin System:

a) Try to use less energy; and

b) Try to reward the miners as equally as possible.

In order to achieve Satoshi's original vision "***one-CPU-one-vote***", i.e. a fully ***decentralized Proof-of-Work, dPoW***, without possibility to form dominant mining pools, we propose a ***public mining pool*** on the blockchain. All full nodes on the network share the public mining pool. Consequently, the public mining pool is separated from individual full node on the network and thus from producing blocks. New block will still be produced by a full node randomly selected from Verifiable Random Function (VRF) and ***Byzantine Agreement (BA)***. This full node uses the public mining pool instead of a private mining pool. The effective hash rate of all mining devices in the public mining pool is much higher because all hash rates work together without repeating the same calculation. The over 51% hash rate attack is also prevented by using public mining pool.



Each miner in the public mining pool is required to apply a public DID_{MINER} using their private key SK_{MINER} and his/her personal online identifier PID_{MINER}. Their DID's VRF hash and proof will be stored on blockchain for the verification of DID_{MINER}. Miner has to choose an address on the blockchain as their public address on blockchain, PA_{MINER}. This will be used to verify their public key PK_{MINER} on the blockchain. On-blockchain public mining pool uses ***Stratum mining pool protocol***。

Similar to ***Algorand*** and ***Dfinity*** blockchains, we use Verifiable Random Function (VRF) and Byzantine Agreement (BA) to randomly select a group of block producers and a group of block verifiers. Similar to miner in the public mining pool, each full node is required to apply a public DID_{NODE} using their private key SK_{NODE} and his/her personal online identifier PID_{NODE}. Their DID's VRF hash and proof will be stored on blockchain for the verification of DID_{NODE}. Full node's address on the blockchain is their public address on blockchain, PA_{NODE}. This will be used to verify their public key PK_{NODE} on the blockchain.

Each block producer will use VRF to select a subset of the public mining pool to perform Proof-of-Work as described below,

a) a subset m of all miners M in the on-blockchain mining pool P to mine the new block b_{new}

- $n \in N$ and $n = \{n_1, n_2, n_3, ……, n_{d-2}, n_{d-1}, n_d\}$ where N is the full set of nodes
- $l = L / d$ where L is the number of miners in M in the pool P and d is the number of nodes in the full set of nodes N
- p is a subset of N to be selected to produce new block
- $m \in M$ and $m = \{m_1, m_2, m_3, ……, m_{l-2}, m_{l-1}, m_l\}$ where l is the number of

miners in the subset m and m_i (1 <= i <= l) is a miner within m

- $node_j$ maps a subset of miners $m^j = \{PK^{m}_{i,j}\}$ in the pool P where 1 <= i <= l and 1 <= j <= d and $PK^{m}_{i,j}$ is the public key of miner i within m^j
- $node_j$ computers a random number $alpha_j$
- $node_j$ computes $beta_j$ = VRF_hash(SK_j, $alpha_j$)
- $node_j$ computes pi_j = VRF_prove(SK_j, $alpha_j$)
- $node_j$ computes $beta_j$ = VRF_proof2hash(pi_j)
- $node_i$ computes VRF_verify(PK_j, $alpha_j$, $beta_j$, pi_j) = 1 where i != j
- a number of $beta_j$ mod pi_j are the selected nodes, i.e. $node^{mod} = \{node^{mod}_1, node^{mod}_2, node^{mod}_3, \ldots\ldots, node^{mod}_{p-2}, node^{mod}_{p-1}, node^{mod}_p\}$ and associated subset of miners m
- a subset of nonce in sequence will be distributed to miner within $node^{mod}$ to perform PoW

b) use Byzantine Agreement (BA) to certify new block

- A subset of nodes $node^{mod} = \{node^{mod}_1, node^{mod}_2, node^{mod}_3, \ldots\ldots, node^{mod}_{p-2}, node^{mod}_{p-1}, node^{mod}_p\}$ are verified and selected to produce new block through the above VRF algorithm
- Each $node^{mod}_q$ has the same set of $\{(beta^{mod}_1, pi^{mod}_1), (beta^{mod}_1, pi^{mod}_1), (beta^{mod}_2, pi^{mod}_3), \ldots\ldots, (beta^{mod}_{p-2}, pi^{mod}_{p-2}), (beta^{mod}_{p-1}, pi^{mod}_{p-1}), (beta^{mod}_p, pi^{mod}_p)\}$ where 1 <= q <= p
- ***Gossip***($header^{new\ block}_q$, PK^{mod}_q, $beta^{mod}_q$, pi^{mod}_q) where 1 <= q <= p
- H_y = Header_verify($header^{new\ block}_z$, PK^{mod}_z, $beta^{mod}_z$, pi^{mod}_z) = 1 where 1 <= z <= p
- H^c_y = Votes_count(H_y)
- H^c_x = Header_certify(H^c_y > 2/3 of p)
- Gossip(H^c_{xy}, PK^{mod}_{xy}, $beta^{mod}_{xy}$, pi^{mod}_{xy}) where 1 <= z <= p
- $H_{new\ block} = Max\{ H^c_{xy1}, H^c_{xy2}, H^c_{xy3}, H^c_{xy4}, H^c_{xy5}\}$ where H^c_{xyv} is the votes received for the block header which is greater than 2/3 of p and 1 <= v <= 5
- Gossip($block^{new\ block}_v$, PK^{mod}_v, $beta^{mod}_v$, pi^{mod}_v) where 1 <= v <= 5
- B_y = Block_verify($block^{new\ block}_v$, PK^{mod}_v, $beta^{mod}_v$, pi^{mod}_v) = 1 where 1 <= v <= 5
- $B_{new\ block} = Max\{ H^c_{xy1}, H^c_{xy2}, H^c_{xy3}, H^c_{xy4}, H^c_{xy5}\}$ where H^c_{xyv} is the votes received for the block header which is greater than 2/3 of p and 1 <= v

If block wins over two third of certification votes and the certifiers representing majority of network nodes, this block will be certified which is final. If a block is not certified which could occur when network is temporarily split, uncertified block with highest accumulated hash rate during the split wins when split networks reunite.

We also allow a ***private mining pool*** to join mining. The private mining pool is also required to apply a miner $DID_{PRIVATE\_MINER}$ and a full node $DID_{PRIVATE\_NODE}$ as well as a $PA_{PRIVATE\_MINER}$ and a $PA_{PRIVATE\_NODE}$. The private mining pool can join the public mining pool or not at will. When it joins the public mining pool, it will be used by all full nodes in the network. This approach has a significant advantage for the private mining pool that the group of producers are selected randomly among all full nodes. Consequently, this private mining pool has much higher probability to mine a new block than it’s attached to a single full node due to the VRF selection process.

We further encourage private mining pool to join the public mining pool through the below algorithm,

nX* + *Y* = *100% of mining award,

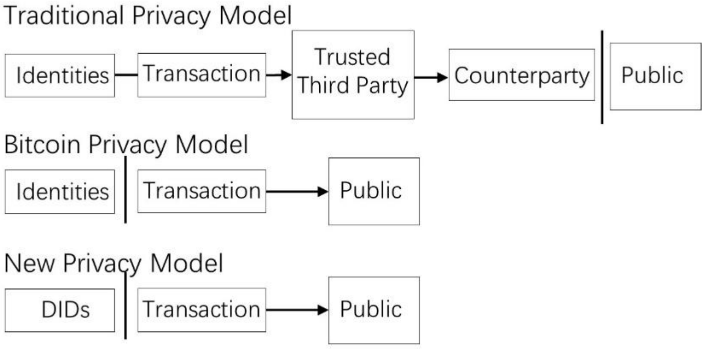
where n is the extra percent award to ***miners X*** in the public mining pool.

On the blockchain, miners in the public mining pool, i.e. miners X, are identified with their $\mathrm{DID}_{\mathrm{MINER}}$ while miners in the private mining pool, i.e. miners Y, are identified with their $\mathrm{DID}_{\mathrm{PRIVATE\_MINER}}$. In addition, miners X can participate community consensus build. It’s important to note that every time community consensus on the value of n is reached, it's necessary to force all nodes and miners to adapt it through a hard fork to prevent different n value being used in the network.

In order to encourage reduce of electricity usage for the benefit of user and society, it’s possible to reduce the difficulty of hash below than the total hash rate of the public mining pool. With lower difficulty, user still has the same probability to mine a block. This will be determined by the community consensus. The change of difficulty of hash requires only soft fork.

6. Privacy

The traditional privacy model relies on a trusted third party and its counterparty. Bitcoin has changed that by using a pair of public and private key and the derived address from the public key. We have proposed a decentralized crypto ID, DID, model to further improve the usability of Bitcoin privacy model. The DID can be verified on the blockchain as user’s crypto identity without user’s direct interaction by sending a transaction to the public address of user, $\mathrm{PA}_{\mathrm{USER}}$. The DID model also allows verification of user’s public key without user’s direct interaction by sending a transaction to the user. This opens a huge door for any third party to implement digital identity application based on the traditional privacy model but without the need of a trusted third party and its counterparty. Since user let the proof or proxy created by VRF to answer crypto identity inquiry instead of using the private key, it does provider stronger privacy protection.



Of course, the new privacy model depends on the security provided by the underlying blockchain and OmniLayer sub transaction. In the algorithm of public key verification, the known public address is used to replace a random input to produce the hash with the private key. This does weaken VRF proof but this is only used for the public key verification not for the digital identity verification.

7. Money Issuance

Money, a special token, may be called ***stablecoin***. It is issued by a smart contract following Money Issuance Protocol, acting the role of a ***central bank***. When a miner gets his ore as reward, he may use it as a collateral to borrow money from the central bank with the following conditions:

a) The collateral ratio ***k*** is set by the Money Issuance Protocol of the central bank;

b) The loan has an annualized interest rate ***r***; and

c) The term is indeterminate as the miner can return the stablecoin any time he wants but the central bank cannot recall the loan.

This way the ore and the stablecoin are separated, which is very different from the case of the Bitcoin System where the ore (Bitcoin) is itself the money.

We assume that the stablecoin is also circulated in the real world --- we do expect a useful monetary system, even if built in a purely virtual world with a peer-to-peer logic, to see its stablecoin floating in the real world. Therefore, there will be an exchange rate ***x*** between the stablecoin and a basket of representative fiat monies in the real world. The exchange rate ***x*** can be imported via an Oracle into the central bank smart contract representing Money Issuance Protocol, which uses it as a variable to drive the interest rate ***r*** and collateral ration ***k***, which in turn will induce more or less miners requesting the stablecoin.

We assume that by adjusting the interest rate ***r*** and collateral ration ***k***, miners can be motivated to borrow more or less, leading to the adjustment of the money supply, resulting in a relatively stable stablecoin.

The specific ways to observe the exchange rate ***x*** can vary depending on the practical situation. The formula connecting x, r and k, is to be designed by each BitTribe as its unique feature.

8. Incentive

Like Bitcoin, mining is still the primary approach to join BitTribe. However, the proposed decentralized Proof-of-Work eliminates those dominant mining pools while decentralized monetary system is designed to reduce significantly the concentration of coins within a few addresses and encourage holding of coins among all miners. We propose these three steps to solidify our commitment. First of all, unlike all other blockchains except a few like Bitcoin and

Bitcoin Cash, the coin of BitTribe is only issued to miner through mining without issuing any premined coin. Second, we proposed a much fair system to mine where dominant mining pool are replaced with a public mining pool. A premium has been placed on the miners in the public mining pool. We use VRF as the random selector to further and intentional to even the play field for small players. Third, the value of each BitTribe coin is not limited to its individual system but linking to all tribes of BitTribe through trade among them. This collective network effect will continuously grow with growth of all tribes of BitTribe. Unlike all other cryptocurrency systems such as Bitcoin, Ethereum, EOS, etc., we preach the BitTribe community to grow like ***a tree of blockchains*** with their own coins instead of a single blockchain with a single coin. Furthermore, other BitTribe community could grow from scratch as a new tree to form a forest of BitTribe communities. The forest of BitTribe communities linked with trade is the true unbeatable value of BitTribe model.

9. Conclusion

Moving from a peer-to-peer electronic cash system to a peer-to-peer monetary system, i.e., from Bitcoin to BitTribe, would be a great leap forward. Although Satoshi Nakamoto did not mention this as a goal in his celebrated paper, we believe that this would be highly consistent with his untold vision. Many challenges lie ahead before we can have a practically useful BitTribe in place. This paper has explored the basic logical elements for a BitTribe and even made some technical suggestions for its implementation. However, the views and proposals in our paper are intended not as conclusive but as inducive, calling for joint efforts from the global community to build one, and then more, meaningful and diverse BitTribe, which, when it happens, will change our world forever.

Bitcoin References

https://bitcoin.org/bitcoin.pdf

Hashrate Distribution amongst the largest mining pools (4 days)
https://www.blockchain.com/en/pools?timespan=4days, 2018

Alireza Beikverdi and JooSeok Song, "Trend of centralization in Bitcoin's distributed network"
https://ieeexplore.ieee.org/document/7176229/, 2015

Lin William Cong, Zhiguo He and Jiasun Li, "Decentralized Mining in Centralized Pools"
http://www.law.nyu.edu/sites/default/files/upload_documents/SSRN-id3160046_1.pdf, 2018

Adem Efe Gencer , Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer, "Decentralization in Bitcoin and Ethereum Networks" https://arxiv.org/pdf/1801.03998.pdf, 2018

Jimmy Song, "Mining Centralization Scenarios"
https://medium.com/@jimmysong/mining-centralization-scenarios-b74102adbd36, 2018

Cryptocurrencies Wealth References

https://coinmarketcap.com/charts/

https://www.ccn.com/86-of-ico-tokens-now-worth-less-than-initial-cryptocurrency-exchange-listing-price/

https://howmuch.net/articles/bitcoin-wealth-distribution

https://www.cryptoglobe.com/latest/2018/08/chainalysis-research-concentrated-ownership-of-bitcoin-cash-bch-responsible-for-low-adoption-in-commerce-just-67-wallets-control-56-of-bch/

VRF & BA References

Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In Proceedings of the 40th IEEE Symposium on Foundations of Computer Science, pages 120–130, 1999

https://datatracker.ietf.org/doc/draft-irtf-cfrg-vrf/

https://datatracker.ietf.org/doc/draft-vcelak-nsec5/

Tibor Jager, "Verifiable Random Functions from Weaker Assumptions", https://eprint.iacr.org/2014/799.pdf

Silvio Micali, Michael Rabiny and Salil Vadhanz, "VerifiableRandom Functions" https://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Pseudo%20Randomness/Verifiable_Random_Functions.pdf

Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, Nickolai Zeldovich, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies"

https://people.csail.mit.edu/nickolai/papers/gilad-algorand-eprint.pdf, 2017

Timo Hanke, Mahnush Movahedi and Dominic Williams, "DFINITY Technology Overview Series Consensus System Rev.1" https://dfinity.org/pdf-viewer/library/dfinity-consensus.pdf

demo of dfinity testnet at https://dfinity.org/, 2018

T-H. Hubert Chan, Rafael Pass, Elaine Shi, "Communication-Efficient Byzantine Agreement without Erasures", https://arxiv.org/abs/1805.03391

DPKI References

https://danubetech.com/download/dpki.pdfhttps://eprint.iacr.org/2014/803.pdf https://www.researchgate.net/publication/318584251_ClaimChain_Decentralized_Public_Key_Infrastructure

OmniLayer Reference

https://github.com/OmniLayer/spec