Academia: A Blockchain-Based Peer-to-Peer Identity System

Ajiansaundo
ajiansaundo@ethmail.cc
upontheewe.eth

David Hopkins
you@empower-humanity.org
4rt.st

Abstract

This paper proposes a decentralized identity system in which cryptographic wallets are verified as being the primary wallet belonging to a real and extant person using the current balance of votes of other verified wallets. This wallet is referred to as a person's signatory wallet and signatory wallet verification is incentivized by a portion of every transaction on the network being automatically distributed evenly across all verified wallets. In addition to casting votes, wallets are used to publish original content to a decentralized and encrypted storage network. Information about a file's origination linking it to the wallet through which it was uploaded is published to a blockchain. As this information is permanent, timestamped and publicly auditable, it may be used in establishing proof of authorship - provided that the author chooses to make access to the published file itself publicly accessible - and, similarly, the opus of original content that a signatory wallet publishes may be used to demonstrate the wallet's legitimacy and deservedness of verification. Alternatively, an author may opt to make published files only accessible privately, in which case the author has the option to make content individually or collectively available to other wallets via subscription or outright purchase. The information regarding which wallets have access to which content at any given time is published to the blockchain and the provision of content that has been purchased is managed by issuing buyers with wallet-dependent (and, optionally, time-limited) private keys that allow them to locate and decrypt the purchased files on the network. In addition to incentivizing wallet verification, the economic activity of the content marketplace is able to subsidize the cost of voting, allowing votes to be cast freely - except for a ubiquitous limit on the rate at which any given verified wallet can issue votes. This limit, which must balance the use of blockspace for financial transactions with its use for votes to ensure that the system remains sustainable, may be decided upon by the majority vote of all verified signatory wallets. All other proposed changes to the system may be decided upon similarly. However, with the exception to votes relating to verification, any vote that a verified signatory wallet casts is done so confidentially (using zero-knowledge proofs). The blockchain is secured using a proof of space-time algorithm that incentivizes network participants to provide computing resources to the system for the storage and retrieval of user data. Files that are published to the network are encrypted and sharded into packets which are distributed at random across the server nodes. Packets are also cloned and these clones are similarly distributed to fill up any available storage space so as to provide as much redundancy as the size of the network permits. Information about the current location and number of clones of each packet is published to the blockchain. In addition to adding and pruning clones as server nodes are added or removed (to retain an equal level of redundancy across all packets), packets are constantly shuffled between nodes in a sequence determined arbitrarily by the algorithm. A log is maintained of each time that a packet is transmitted across the network and a new block is forged for every nth successful transmission, rewarding the forger that owns the node from which the packet was retrieved. Owners of server nodes can also elect to flexibly partition the server space that they are making available to the network to create a vault for storing whole files. This storage real estate may then be rented on a marketplace, so that users may have the option of storing files on the network where they will be served from a location that is proximate to them, and for which server node operators' reputations may be established by reviewing their block forging history.

1 Introduction

We propose Academia as a decentralized identity system. In this system, secure signatory wallets are used to publish digital content to a distributed network [1]. A unique, timestamped signature is created for each item of content that is published, associating it with a signatory wallet, and each

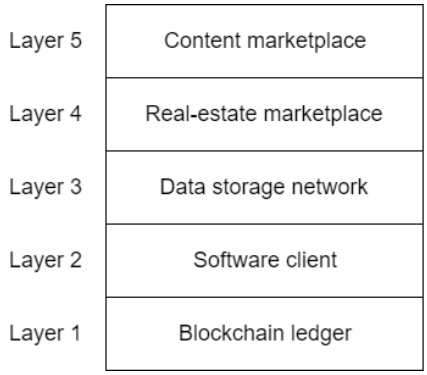

Figure 1: Academia consists of multiple layers: *Layer 5* involves the sale and purchase of content uploaded to the network. *Layer 4* consists of the use of storage vaults, which allow files on the network to be locally stored, and the marketplace through which storage vaults can be rented. *Layer 3* consists of the decentralized network upon which data is sent and received and may be stored permanently, where block forging also occurs. *Layer 2* is the common software that is used to read the blockchain ledger and determines what may be added to it. Changes to this layer may only be made by the majority vote of participants verified by the network. *Layer 1* is the historical record itself.

signature is also published to the blockchain. The opus of original digital content that a wallet has signed may then be used as proof of the wallet's authenticity as being the primary signatory wallet of a real, extant, unique person by other participants in the network [2].

Any verified signatory wallet can choose to cast a vote on the legitimacy of any other wallet. An issuer's vote can be changed at any time by issuing a new vote and votes are recorded to the blockchain. The current balance of votes towards a wallet by other verified wallets determines whether or not it is considered verified and so may vote on other wallets, with the exception of a ceremonial vote that is applied temporarily using an inception signatory wallet in order to ignite the voting system.

Items of digital content that are published to the network may be made publicly or privately available and, when private, access to them may be bought and sold individually or collectively to the owners of other signatory wallets either permanently, by subscription or both.

2 Governance

The transactions conducted by the marketplace subsidize the cost of publishing votes to the blockchain, which are always made free to cast. As such, the rate at which votes may be cast is limited by the amount of economic activity that is occurring on the network; as blockspace is limited, the number of votes that each new block may include is inverse to the number of transactions that it may include.

Decisions regarding updating the system - and in particular the amount of blockspace that is allocated to transcribing votes (which sets the rate at which any given verified wallet may cast votes) - are made by a majority vote of all verified signatory wallets.

Data that is published to the network is encrypted, sharded and distributed across the network of nodes offering storage space. Shards are duplicated to expand to fill the storage space that is available, such that the reliability of data permanence is related to the amount of data uploaded to the network versus the amount of storage space that is made available.

Storage space may be offered to the network by downloading a software client and allocated drive space to be populated by it, which is incentivized by the block forging rewards which can be earned as a result. The amount of space that a user allocates ot the network may also be managed flexibly, such that space that is not taken up with one's own files is automatically assigned to the network.

3 Block Forging

Block forging uses a variant of proof-of-spacetime [3] that constantly tests the speed of data retrieval across the entire network of server nodes, rewarding each owner of a server that executes every nth

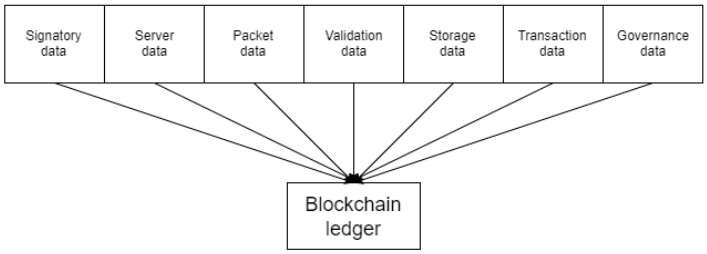

Figure 2: The ledger contains multiple types of information within it, including: (a) the votes that wallets have cast relating to each others' verification, (b) the historical order in which server slots were made available to the network and which of these slots are no longer active, (c) the current and recent location of each packet on the network, (d) the rate at which tokens are to be forged and the number of successful packet transmissions since the last token was forged, (e) information relating to the management and rental of data storage vaults, (f) transaction data, and particularly relating to the sale and purchase of access to files on the network and (g) a permanent link [4] to a copy of the core software client that is used by the network and voting results related to its proposed changes.

successful test, with the rate of change of n increasing over time.

In addition to executing tests, a portion of drive space used by the software client may also be allocated to an encrypted vault in which whole files may be stored (and this allocation may be flexible). Access to this vault may be kept or rented on a public exchange - where storage providers may be evaluated based on their history of successfully executing the block forging algorithm. This allows users to find proximate locations where one's own files may be accessed so that may be retrieved at greater speed (including one's own device) while continuing to make them available to be traded on the network.

Ten percent of every transaction that is conducted on the network is evenly distributed amongst all of its verified signatory wallets. This ensures that ownership of a verified wallet is valuable and encourages participation in the network [2].

4 File Storage

User A submits a file (F) to be published to the network using a software client to which the user has signed into with a signatory wallet.

A hash of the signatory wallet, the file and the timestamp at which it is uploaded [5] is published to the blockchain.

File F is encrypted using a private key and split into shards, each of which is size N (for example, 1mb), with padding used on the last shard.

Each of these shards is assigned an ID that may be validated using a mod10 algorithm (or similar) [6].

The ID of each shard is then encrypted and each shard is appended with its encrypted ID. Each shard is also appended with a list of all of the other encrypted IDs of shards for that file.

The origin of the shards is anonymized using a confidentiality algorithm - such as a non-interactive zero-knowledge proof [7].

A constant measure is maintained by the network of the number of shards that it stores versus its storage capacity. This information is used to determine the ideal number of times to clone each shard. The shards and their clones can then be distributed randomly across the servers on the network in the form of packets. A measure of the number of clones of each packet is maintained on the blockchain and clones are automatically pruned or added and then (quasi-)randomly distributed to ensure and even amount of data redundancy across all packets and based on the current storage capacity of the network. This activity is governed by the forging algorithm described in Section 6.

When User A seeks to retrieve file F, the user polls the network and as soon as a shard is identified that returns a valid ID when decrypted using the private key of the user, it yields the ID of all of its sibling shards. This allows all of the files shards to be quickly located on the network for downloading by the user. As there are multiple clones of any shard, some of which may be more proximate to the

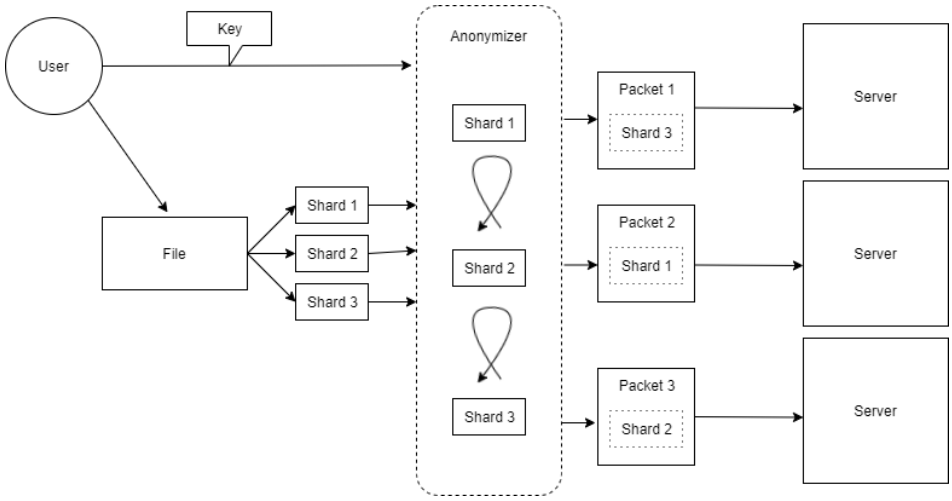

Figure 3: Files are encrypted and split into equal-sized shards, their identity is made confidential and they are randomly distributed across the server nodes.

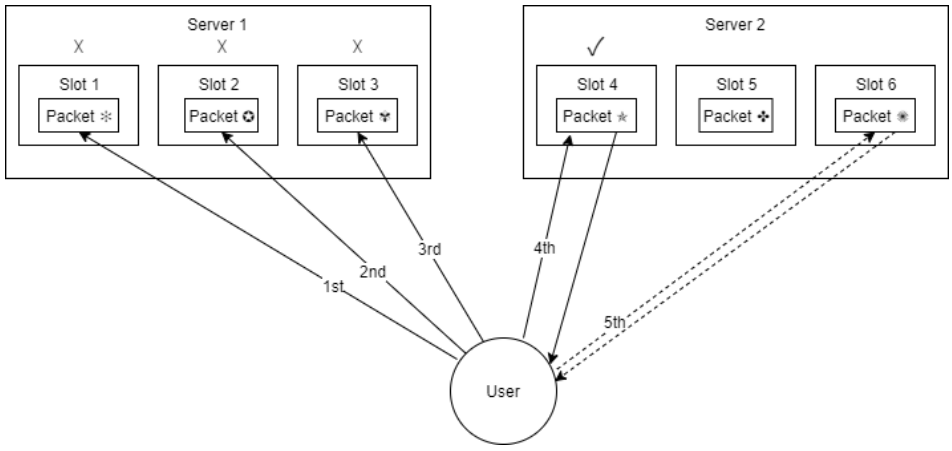

Figure 4: To download a file, the user polls the network with a decryption key. The first packet that is identified as having an ID that is decipherable using the key is downloaded (and a second key is used to decrypt its contents locally). In addition to its file shard, each packet contains an encrypted list of the IDs of the remaining packets that are required to reconstitute the file, allowing the remainder of the file to be located quickly.

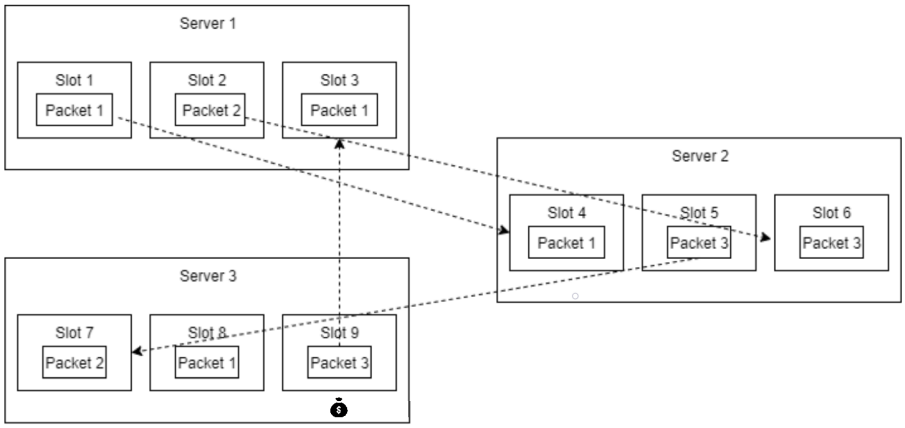

Figure 5: Packets are constantly being shuffled around the network in a sequence determined by the blockchain algorithm, with a reward distributed to the donor node of every nth successful transmission.

user than others, this process may be made faster through the use of technologies such as torrenting [8], to which it is closely related.

The process of verifying whether a transmission was successful is made more robust by the introduction of intermediary nodes. The difficulty of falsifying a successful transmission increases as the number of random nodes that a transmission is plotted through expands.

If necessary, it is also possible to provide extra obfuscation against surveillance of network traffic (to determine which shards are being downloaded in attempt to reconstruct the encrypted file, lest an attack should be made to target those servers in particular), such as by downloading a wide array of shards at random from the network in addition to those that belong to the file and then deleting the extraneous shards locally.

5 Proofing Algorithm

Forgers offer computing resources to the network and, in particular, dedicate storage space, processing power and bandwidth. The device must be powered on and connected to the internet in order to yield rewards and is assigned a unique ID that associates with a wallet owned by the forger. In addition to the space allocated for file storage, a copy of the blockchain itself is stored on the forger's computer. Certain datum that are stored on the blockchain (described below) are not required to be stored indefinitely.

The space on the forger's computer that has been made available to the network is populated by small, equally-sized packets with a unique identifier which are downloaded to the forger's device. Each of these packets contains an encrypted file shard (as described in the previous section). When a new server is added, these packets will be clones of other packets already on the network. Which packets are cloned and downloaded is determined by which packets currently have the least clones available so that the number of clones of each packet always remains constant.

The forger's device constantly polls the network to determine how many clones of each packet exist on it and the extent to which this number deviates from the average number of clones per packet so that the local packets may be constantly pruned and replaced to ensure that the maximum number of clones is kept equal for every packet on the network (which protects data permanence).

In addition to this, packets are "arbitrarily" moved between devices. The blockchain contains a record of the hash of each packet and the ID of each device on which a clone of it is currently being stored. Processor speed that has been allocated to the network by a node device is used to "arbitrarily" replace local packets with foreign ones and update this data to the network.

Each of these transmissions of packets - whether it be to balance the number of clones of a packet, for reshuffling or both - causes a counter to be incremented. Downloads of packets across the network by users seeking to reconstitute files are marked separately, and should always be accompanied by

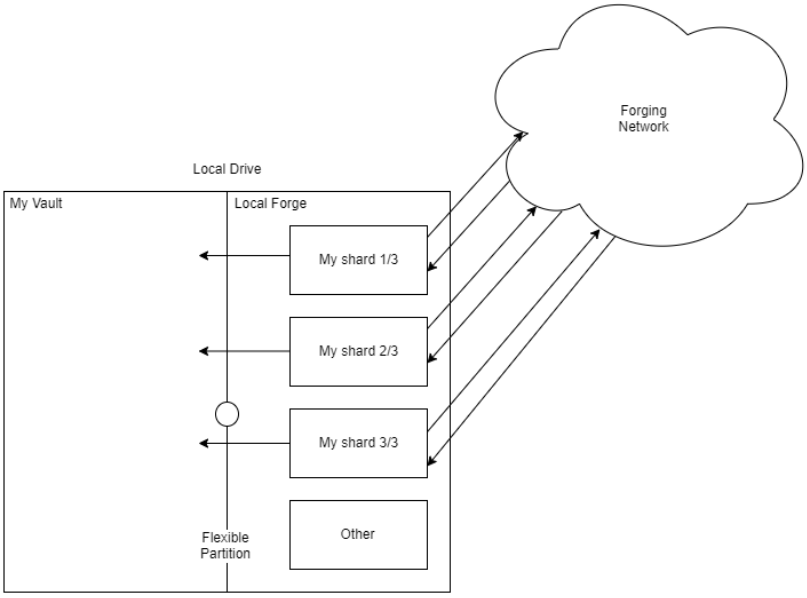

Figure 6: A local drive can have a vault (for whole files) and a forge (which contributes space to the packet network). Any empty vault space can be allocated to forging and forging space can automatically start to empty when the vault overflows. Provided that the activity can be obfuscated sufficiently as necessary, this Figure also demonstrates how packet reshuffling may also be used to download files: one simply replaces local slots with the packets of the requested file and then shifts them over into the vault to be combined.

sufficient "arbitrary" activity to prevent packets from being predictably located on any given device (which could be exploited)

The state of the counter, a hash of each packet that is transmitted across the network (which can be used to validate that a packet has not been corrupted prior to re-transmission) and the device IDs of the origination and destination nodes involved are all recorded to the blockchain. While information such as the hash of all packets on the network and their current locations is important, the record of a packet's previous locations become increasingly less important over time and so it is foreseeable that some of this information may be able to be cleaved from the copy of the blockchain that client nodes store locally in order to conserve storage [8].

A block is forged for every nth packet that is successfully retrieved on the network by any other node and tokens are granted to the owner of the node from which the packet was retrieved.

If a transaction fails and a packet is not retrieved successfully, that information is also published to the blockchain so that that packet may now be considered dead and the count of the number of clones of that packet that are available on the network be adjusted accordingly. When a packet on a node has been declared dead, it will be marked for clearance on that node so that it may be replaced with a live packet.

Packets cannot be re-added to the network once they have been declared dead and the only packets on a server node that are recognized by the blockchain are those that have been added using the blockchains algorithm (and recorded to it).

6 Validation

An auditable process whereby it may be determined algorithmically which packet is to be transmitted across the network next - and by using which nodes - may be implemented as follows.

Server space is divided into slots, with each slot equalling the size of a packet on the network. Slots are recorded on the blockchain in the order that they are added to the network, and an on-chain record is also kept of whenever a slot fails to receive a packet (so that it may be marked as dead). In addition to the information listed in Section 5, transaction records contain the IDs of the donor and

donee slots, which may be used to determine comparatively how long ago (a) each slot has donated a packet and (b) had its own contents replaced.

The slot that has remained unmodified for the longest time that contains a packet with equal to the most number of clones is emptied. The contents of the slot are then filled by a transmission from the node carrying the slot containing a packet that has equal to the least number of clones on the network that has not donated a packet for the longest time.

This algorithm may be used complementarily to requests by users to download data that has been uploaded to the network. In particular, a certain number of algorithmically-determined movements are required to occur per every user-initiated transaction to ensure the health of the system.

As the sequence whereby these transmissions must occur in order to be added to the blockchain is publicly verifiable [10] and transmissions are evenly distributed across the network, network participants may have confidence that rewards will be distributed fairly. Moreover, the system incentivizes activity that will provide greater reliability and speed to its users and the long-term optimization of data storage and data retrieval technology.

7 Primitive

A simplified version of Academia may be implemented as a protocol using existing blockchains. The main disadvantages of doing so relate to network fragmentation - there are weaker incentives for the network to grow and for a single network to become ubiquitous - and there being less substantial defences against the system being corrupted over time as a result.

Nevertheless, to elucidate the process of wallet verification, we examine the following example of how it may be achieved rudimentarily.

A community decides on an *inception wallet*, a *positive vote wallet*, a *negative vote wallet* and a standard method whereby content is hashed (or a standard method whereby a permaweb link is published to the blockchain [4]). (Third-party tools are also important for collecting the information that is published to the blockchain together and presenting it in a succinct format so that it may be parsed quickly.) It is immaterial which blockchain is chosen provided that it is publicly auditable, although ideally whichever is used has low transaction fees.

When a wallet submits a transaction to the *positive vote wallet*, the next wallet that it submits a transaction to constitutes an affirmative vote as to whether that wallet should be verified. Similarly, when a wallet submits a transaction to the *negative vote wallet*, the next wallet that it submits a transaction to constitutes a negative vote as to whether that wallet should be verified.

The current tally of all votes from verified wallets of a given wallet indicates whether it is verified. (With the exception of the first vote that is given by the *inception wallet*, which is counted even though the *inception wallet* has not itself been "verified" yet. The second vote is then cast by the wallet verified by the *inception wallet* to verify the *inception wallet* and legitimize the casting of the first vote, after which all subsequent votes may proceed in the standard manner for the lifetime of the system. A similar result may be achieved by allowing the *inception wallet* to cast as many votes as desired initially and then becoming considered a regular wallet once either a notionally verified wallet casts a vote to verify the *inception wallet* or any two notionally verified wallets vote to verify each other.)

If a person wishes to establish their authorship of an item of content, they may upload it to the permaweb and then use whatever process has been determined to use their signatory wallet to publish a link to the content to the blockchain. Alternatively, the person may simply use a process that associates a hash of the content to a particular receiving wallet address and then make a transaction from the person's signatory wallet to that address. (This method can also be used in isolation of a proof-of-identity system purely to support claims of authorship.)

8 Conclusion

We have proposed a blockchain-based content marketplace for peer-to-peer identity verification. The marketplace associates a signing wallet with the opus of time-stamped, original content that it has signed, using the current balance of votes from other verified wallets to determine it to be the supreme wallet of a real, extant person. The published content not only serves as proof of the individual but

access to it can be bought and sold. As long as financial transactions continue to subsidise votes being freely published by verified wallets to the public ledger, changes to the codebase are also able to be made democratically. Data stored on the network is sharded and distributed.

In order for the initial wallet to become verified, a temporary vote may be considered to be assigned to it that does not originate from any wallet, allowing it to cast votes that may be considered notionally legitimate until a pair of mutually-verifying wallets are established, at which point the special permission granted to the initial wallet rescinded automatically.

The system's security is maintained by a decentralized network of server nodes that contribute computing resources in exchange for tokens that may be used as currency in the marketplace. Packets of data on the network are constantly shuffled between nodes using a pre-determined algorithm which improves the overall reliability of the network as a data store and rewards node size and data retrievability. This algorithm, which combines elements of proof-of-spacetime and proof-of-work, may also be referred to as a proof-of-God algorithm [11].

References

[1] S. Wilkinson, "Storj: A Peer-to-Peer Cloud Storage Network," https://www.storj.io/storj2014.pdf, 2014.

[2] D. Hopkins, "Canceling in the Moonlight: Power in the Age of Catechism," Preprint at https://medium.com/demliberxls/canceling-in-the-moonlight-power-in-the-age-of-catechism-draft-74cc12aaf9bb, 2021.

[3] T. Moran, I. Orlov, "Simple Proofs of Spacetime and Rational Proofs of Storage," https://eprint.iacr.org/2016/035.pdf, 2016.

[4] S. Williams, W. Jones, "Archain: An Open, Irrevocable, Unforgeable and Uncensorable Archive for the Internet," https://www.arweave.org/whitepaper.pdf, 2017.

[5] National Institute of Standards and Technology, "FIPS 180-2 with Change Notice 1," https://csrc.nist.gov/csrc/media/publications/fips/180/2/archive/2002-08-01/documents/fips180-2withchangenotice.pdf, 2002.

[6] H.P. Luhn, "Computer for verifying numbers," U.S. Patent 2950048A, 1960.

[7] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, "Zerocash: Decentralized Anonymous Payments from Bitcoin," In *2014 IEEE Symposium on Security and Privacy*, 2014.

[8] B. Cohen, "Incentives build robustness in BitTorrent," In *Workshop on Economics of Peer-to-Peer Systems*, vol 6, June 2003.

[9] V. Buterin, "Endgame," https://vitalik.eth/general/2021/12/06/endgame.html, 2021.

[10] S. Nakomoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf, 2009.

[11]The Bible, King James Version.