Durabit: A Bitcoin-native Incentive Mechanism for Data Distribution

someguy
someguy@durabit.org
www.durabit.org

4de67a207019fd4d855ef0a188b4519c7040281c9c121fc28574b0bd58bd3927
hash@durabit.org
www.durabit.org

Abstract. Bitcoin can be used to create long term incentive structures for the continual seeding of large files via the use of bitcoin bonds and timestamping torrent magnet links. Durabit introduces the concept of time-locked bitcoin transactions forming a bond that rewards users for their active participation in the seeding of specific files, all while ensuring immutable data verification through cryptographic hashes. Durabit aims to incentivize both the initial data distribution and offset the associated long term operational costs of seeding files.

1. Introduction

The proliferation of digital content, particularly in the realm of peer-to-peer file sharing, necessitates a dynamic mechanism to motivate users to actively seed data. Durabit seeks to address this challenge by introducing a unique incentive system built on the foundation of bittorrent client magnet links, the immutable publishing properties of Bitcoin's blockchain, and time-locked bitcoin transactions to pay out rewards verifiably over time

2. Magnet Links and BitTorrent

When a user decides to download a file using their BitTorrent[1] client, they are generally presented with two possible methods for connecting to peers on the network: Torrent files or magnet links. It is important to note that BitTorrent clients do not engage in the direct downloading of files from the tracker within the magnet link or torrent file. The role of the tracker within the torrent ecosystem is solely confined to monitoring and managing the connections between BitTorrent clients within the peer-to-peer network, devoid of any direct involvement in the downloading or uploading of data.

A decentralized and "trackerless" torrent system is characterized by its capacity to enable BitTorrent clients to establish direct communication without reliance on centralized servers. This functionality is achieved through the utilization of distributed hash table (DHT) technology, wherein each BitTorrent client operates as an autonomous DHT node. When a torrent is added through a "magnet link," the respective DHT node initiates contact with neighboring nodes, and this process cascades through a network of peers until the requisite information regarding the torrent is obtained. This configuration essentially transforms every peer into a functional tracker, thereby obviating the need for a central server to administer a torrent swarm. Consequently, BitTorrent evolves into a fully decentralized peer-to-peer file transfer system. It is noteworthy that DHT can coexist with conventional trackers. For instance, a torrent can simultaneously employ both DHT and a traditional tracker, affording a redundancy mechanism in the event of tracker failure.

Torrents based on magnet link hashes exhibit remarkable resilience. As long as at least one seeder is active, anyone with the magnet link can locate them, even if none of the original seeders are available. If the hash can be regenerated from the torrent files, then existing magnet links will continue to function effectively. A cryptographic hash refers to a mathematical algorithm used to process data, yielding a concise and distinct character string representing the data itself.

This property becomes advantageous in the context of torrents, as it allows for the comparison of the data of any two active peers by examining their respective hashes. When two peers distribute the same files, they will yield identical hash values. Consequently, in the context of a torrent client's operation, identifying valid participants from a particular magnet link becomes a straightforward process. By comparing the hash present in the magnet link to the hashes of the shared torrents, the client efficiently assembles a network of peers by retaining only those seeding files with matching hashes.

Magnet links offer a distinct advantage over torrent files due to the small size derived from their text-based nature, especially when considering the cost per byte when publishing within a Bitcoin block.

3. Timestamping

Timestamping presents a mechanism for proving the existence of a specific piece of data at the time a Bitcoin block is mined committing to that data. It also inherently provides an integrity check mechanism when in possession of a copy of the original data by means of comparing against the original hash that was timestamped.

Both of these properties can be leveraged in the Durabit scheme by publishing magnet links to the blockchain using OP_RETURN. This allows participants to verify the association between a specific torrent file and a bond associated with it, as well as ensure the file they are seeding is the exact file the bond was funded for.

4. Bond Design

The primary issue within the current torrent ecosystem is the incentive for a user to continue to seed after the file has been downloaded in full. There are surely costs associated with long term seeding of torrents, such as electricity usage, broadband demands, and even the data storage itself. By using time-locked

Bitcoin[2] transactions, data brokers can establish a public payment structure to incentivize the long term seeding of specified magnet links. The design facilitates separating the roles of funding the incentive and monitoring and processing payouts to seeders.

4.1 Pre-Signed Bond Transactions

The structure of these transactions requires two participants, the issuer of the bond providing the initial funding, and a Chaumian ecash mint operator to act as recipient of payouts and distribute them to torrent seeders. After the initial establishment of the bond and providing the pre-signed transactions to the mint operator, communication required from the issuer to the mint is unidirectional and minimal. In the simplest implementation none is required at all.

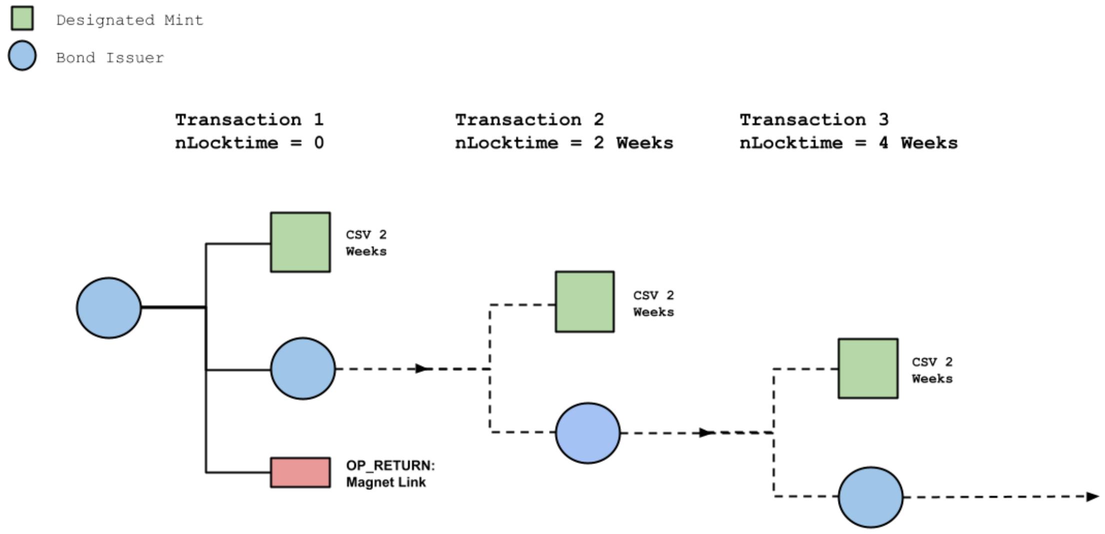

Figure 1. The basic bond structure composed of presigned nLocktime restricted transactions from the issuer. Each transaction creates an output paying to the designated mint with a CSV relative timelock, and a change output returning the rest of the bond funds to the issuers control. The first one additionally publishes the Magnet Link via OP_RETURN. The issuer constructs the transaction series and provides them to the designated mint. The remaining balance of the bond in the issuer’s control can be revoked at any time.

An initial funding transaction allocates a CSV time-locked output to the designated mint, returns the rest of the funds to the issuer’s control, and uses OP_RETURN to encode the magnet link of the torrent the bond is funding the distribution of. Each successive transaction creates a CSV restricted output for the mint, and one that returns the remaining funds to the issuer's control. This guarantees that each payout to the mint is only spendable after the maturity period, enabling the issuer to monitor their behavior. If the mint is not distributing funds to torrent seeders honestly, the issuer can revoke the remaining bond payouts at any time by double spending the output under their control funding them.

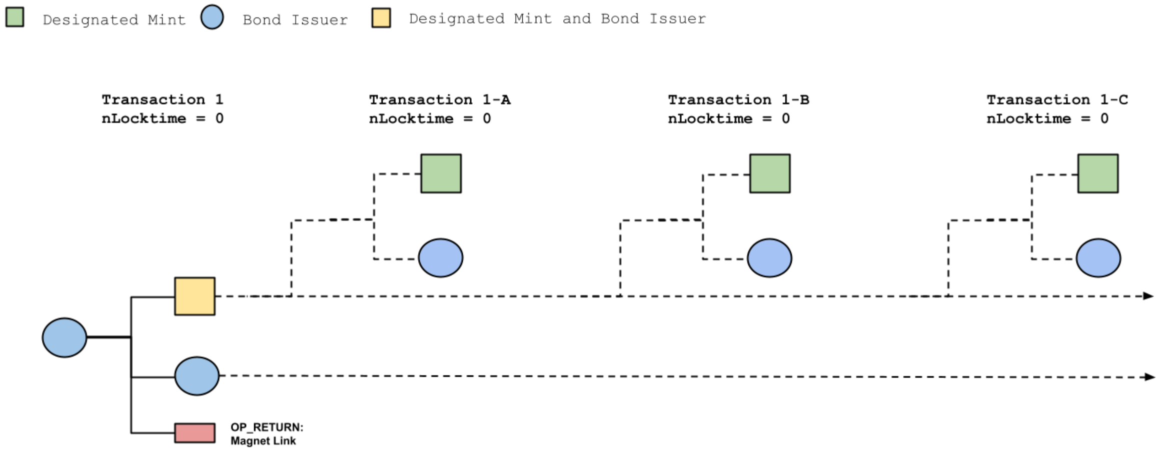

Figure 2. If a broadcast channel is available to the bond issuer that the designated mint can monitor, and the mint makes a public key it controls the corresponding private key to available publicly, each CSV relative timelocked output in the bond transactions from Figure 1 can be replaced with a Spillman channel. The issuer can then periodically broadcast new signatures for transactions allocating more funds to the designated mint. This allows the issuer more granularity in exercising revocation of the bond.

If the issuer can maintain a system capable of periodic broadcasts using a channel the mint can monitor, each CSV restricted output in the first variant of the scheme can be replaced with a 2-of-2 multisig funding a modified Spillman payment channel[3]. This would allow the issuer to progressively unlock more of the funds from each distribution output by broadcasting its signature on a transaction allocating more funds to the mint. This would give the issuer more granular control over revoking the bond. While the issuer would be unable to close the channel due to only having its half of the required signatures, even a dishonest mint is incentivized to close the channel in order to claim the funds allocated to them by the issuer. Because of this incentive and concerns with uncertainty regarding future fee rates and confirmation times, no time-locked refund transaction returning the channel funds to the creator is used.

4.2 Chaumian Mint and Seeders

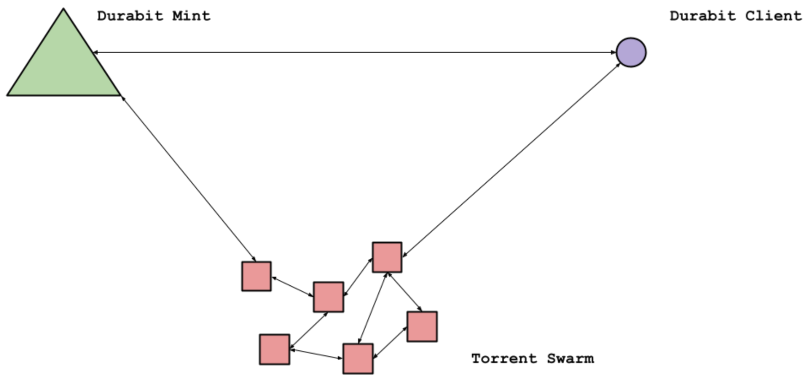

Figure 3. A diagram of the network relationships between the mint and clients seeding a bonded file. The mint interacts with the relevant torrent swarm in order to verify clients that connect to it are actually seeding. It does so by comparing the network address identifier a client connects from to active seeders in the torrent swarm.

The mint, as the recipient of bond payouts, plays the role of distributing funds to users seeding the bonded torrent file. To do so it monitors the torrent swarm of the file in question, tracking users who actively seed. Users running a Durabit client can connect to the mint from the network address they are seeding the file from and register to receive Chaumian tokens issued against the bond payouts.

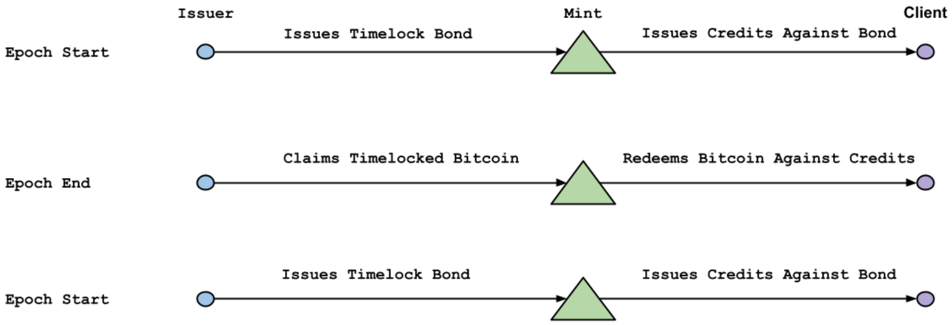
Issuer
Mint
Client
Epoch Start
Issues Timelock Bond
Issues Credits Against Bond
Epoch End
Claims Timelocked Bitcoin
Redeems Bitcoin Against Credits
Epoch Start
Issues Timelock Bond
Issues Credits Against Bond

Figure 4. Global epochs are defined as the interval between the expiry of timelocks in any two bond transactions from Figure 1. Within each global epoch, according to its distribution mechanism and schedule, the mint shall distribute chaumian tokens representing a claim against the yet to be distributed bond payout amongst active seeders. At the end of each global epoch they shall redeem them for bitcoin.

All registered seeders will earn tokens according to the mint's distribution scheme during each epoch — the period between the mint's latest output being created and it becoming spendable — and be able to redeem them at the end of each epoch. The distribution scheme is left entirely up to the mint. The two most obvious schemes would be proportional to bandwidth consumed seeding, or a randomized lottery amongst seeders to mitigate the risk of sybiling a swarm with multiple redundant network identities. In the case of the latter, Bitcoin block hashes could be used as a source of entropy for a Verifiable Random Function (VRF) to select lottery winners. The percent of each bond payout the mint keeps for itself each epoch is also left up to the mint. If an issuer finds the percentage too high, they can choose to not issue a bond with a particular mint.

The mint at any time can challenge any individual seeder to provide hashes of arbitrary pieces of the torrent file, using latency as a measurement to detect on the fly retrieval and relay from another source without hosting the data themselves. In addition to this, requiring hashing the torrent chunk with a recent blockhash as part of the challenge process could add additional assurances of recent possession of the data.

4.3 Bond Distribution Curve

The Durabit protocol escalates the bounty payouts to provide a sustained incentive for data seeding. This is done not by increasing rewards in satoshi terms, but rather by increasing the epoch length between payouts exponentially, leveraging the assumed long term price increase due to deflationary economic policy in order to keep initial distribution costs low.

For example: A torrent link is published with 1 bitcoin total in rewards. The first time-locked transaction allocates 0.1 bitcoin. Subsequent transactions increase the duration between payouts exponentially, with the current duration multiplier determined by the number of difficulty adjustments, or DAs:

0.1 bitcoin - 1 DA (Approximately 2 Weeks)
0.1 bitcoin - 3 DAs (Approximately 1.5 Months)
0.1 bitcoin - 9 DAs (Approximately 4 Months)
0.1 bitcoin - 27 DAs (Approximately 1 Year)
0.1 bitcoin - 81 DAs (Approximately 3 Years)
0.1 bitcoin - 243 DAs (Approximately 9 Years)
0.1 bitcoin - 729 DAs (Approximately 28 Years)
0.1 bitcoin - 2,187 DAs (Approximately 84 Years)
0.1 bitcoin - 6,561 DAs (Approximately 253 Years)
0.1 bitcoin - 19,683 DAs (Approximately 757 Years)

4.4 Incentive Alignment

This design assumes the motivation for an issuer funding a bond is to incentivize the propagation of the bonded data. Therefore it is not within the issuer’s interest to revoke a bond unless they no longer need the information propagating, or are responding to a dishonest mint to reissue the bond with a different mint. If they do however revoke the bond for any arbitrary reason at all, the mint will still be guaranteed the payout at the end of the current epoch. This will allow them to pay out seeders for that epoch. If a mint is ever dishonest in its payouts to seeders, they are only capable of defrauding the issuer for a single epoch’s payout, after which the revoked bond transactions will no longer be submittable on chain.

The bond issuer can periodically at random intervals monitor the torrent swarm for the file they have bonded. Any drastic decrease in seeding activity could be a sign of malicious behavior on the mint's part. The issuer could immediately revoke the bond and reissue with another mint, or directly start seeding the torrent themselves to verify whether the mint was paying out the bond honestly. Revocation could be done following verification they are not paying out honestly.

This achieves a well-balanced incentive alignment that is robust against single points of failure, and can be made more so in the future with extensions to the design.

5. Implementation

The ideal implementation would be a forked custom Cashu[4] mint implementation and Cashu client implementation both with integrated BitTorrent clients.

The Durabit Cashu mint implementation would maintain external dependencies for its Lightning node and Bitcoin full node requirements and use the existing Cashu protocol, simply implementing the internal logic necessary to engage in issuance of tokens conditionally based on its view into the torrent swarm(s).

The Durabit Cashu client would use the existing Cashu protocol and maintain dependence on an external Lightning wallet for withdrawals from the mint, implementing the logic to register with the appropriate mints for torrent files it is seeding with a corresponding bond. Support for calling the Bitcoin Core API or an Electrum server API would allow for independently verifying bonds issuance on chain, and enable future extensions making use of OP_RETURN.

Support for constructing, signing, and revoking bond transactions could be implemented in the Durabit Cashu client, or as a standalone tool.

6. Future Extensions

A number of possible extensions that could be made to the protocol:

1. Include data needed to contact a designated mint managing a particular bond payout in the first transaction along with the magnet data. This would allow Durabit clients to locate and contact mints to begin seeding simply by scanning the blockchain.
2. Having federated mints. This could mitigate the need for a bond issuer to have to revoke and reissue a bond in response to mint failure, disappearance, or dishonest behavior.
3. Federated mints with decaying timelocks, i.e. progressing from a 3 of 5 to a 2 of 5, etc. This could further mitigate the risk of liveness failures that would require revoking and reissuing the bond.
4. Crowdfunding the bond and composing a multisig of funders to sign the bond transactions and control revocation authority.

5. Decrementing the value of individual bond payouts as epochs lengthen to mitigate the risk of a mint's incentive to steal a large pot of individual bond payouts that have substantially increased in value. I.e. instead of lengthening epochs, reduce the value in bitcoin terms of each epoch payout.

7. Conclusion

Durabit aims to incentivize durable, large data distribution in the information age. Through time-locked bitcoin transactions and the use of magnet links published directly within Bitcoin blocks, Durabit encourages active long term seeding and helps offsets operational costs. As the bounty escalates, it becomes increasingly attractive for users to participate, creating a self-sustaining incentive structure for content distribution. Durabit has the potential to architect a specific type of information goods market via monetized file sharing and further integrate Bitcoin into the decades long peer-to-peer revolution.

References

1. Bram Cohen, "Incentives Build Robustness in BitTorrent," (May 22, 2003), bram@bitconjurer.org.
2. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," (October 31, 2008), https://bitcoin.org/bitcoin.pdf.
3. Jeremy Spilman, "Anti DoS for tx replacement," Bitcoin Development Mailing List, April 20, 2013, https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-April/002433.html.
4. Calle, “Cashu: Chaumian ecash wallet and mint for Bitcoin,” (September 23, 2022), https://github.com/cashubtc/nuts