SpOCChain Whitepaper

Validated & Trustworthy Distributed Space Object Catalog Generation Using Blockchains

Dr Stuart Grey

August 30, 2018

Abstract

A purely peer to peer version of the space object catalogue would allow the rapid determination and dissemination of space objects without relying on a single controlling entity. The use of a distributed ledger, recording observations and associated orbits in a cryptographically verified blockchain is proposed. An entirely open ledger must operate in a trustless environment and a system of orbit determination as proof of work and digital signatures offer a solution to this problem while also ensuring that the computational work required to verify the blockchain offers real utility to the community relying on it.

We propose a solution to the determination and dissemination of RSO tracking data and orbits using a peer-to-peer network and associated distributed ledger. The network timestamps observations and orbits by hashing them into an ongoing chain of orbit determination based proof of work, forming a record that cannot be changed without redoing the proof of work.

This chain, or distributed ledger, is open and available to all for utilisation, verification and re-processing.

The immutability of the observation data and predictions on the chain allows for the allocation of tokens to those nodes contributing the most accurate predictions and the underlying observations. The longest chain not only serves as proof of the sequence of observations and predictions but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they will generate the longest chain and outpace attackers.

The network itself requires minimal structure. Observations and predictions are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1 Introduction

The advent of cryptocurrencies, alternatives to fiat currency based on the idea of a distributed ledger verifiied by proof of work, pioneered by BitCoin, has led to proliferation blockchain based products in areas where a public record and verifiablilty are required. At a high level, Bitcoin and other cryptocurrency/blockchain technologies offer have the following attributes.

- Powerful decentralised system (computationally and conceptually)
- System for coordinating resources on a massive scale
- Embedding trust into a network-centric system of coordination
- Full transparency and full anonymity
- Agreed data exchange protocol - ideal for autonomous agents, get humans out the loop.
- Distributed Ledger
- Proof of Work

All of these traits are immediately highly attractive from a space traffic management data sharing perspective.

Two of the key issues facing the SSA community, the inability to share observations and the lack of a shared, validated catalogue, hold back progress in a wide range of areas from operations to fundamental research.

Two technologies, a cryptographically secure distributed ledger and the idea of 'Useful Proof of Work' offer an avenue to tackle these problems in an elegant, scalable manner. The recent popularity of cryptocurrencies and 'blockchain' technologies in general belies the fact that nearly all of technologies involved in these system were in fact developed and in the 1980's and 1990's. The amount of academic effort and recent development in these technologies however offers a real boon to the SSA problem by developing these fundamental concepts in new directions ideal for tackling the very specialist problems in SSA.

This paper, we propose the foundational components of an open, trustless, space object catalogue based on a cryptographically verified distributed ledger and processed using useful orbit determination algorithms rather than repeated cryptographic hashing.

2 Cryptographically Secure Distributed Ledger

In Bitcoin the full transaction history of the currency is recorded in a trusted and cryptographically verified ledger. If we expand our ideas of 'transactions' beyond payments into the realm of STM then we could envisage the exchange of:

- tracking data
- orbit determination
- object disambiguation
- orbit prediction/retrodiction
- course conjunction analysis

The issues with sharing observations and the dissemination of a shared catalogue all come down to trust. In the ledger proposed by Haber and Stornetta [HS91] 'documents' are constantly being created and broadcast (the idea of a document in this context is very general and can in fact be any type of data). In this project the documents will be observations and determined orbits of space resident objects.

In order to build trust in the ledger, as an observation or orbit is added the creator asserts a time of creation and cryptographically signs the observation/orbit, its timestamp and the previously broadcast orbit/observation. This means that as observations/orbits are added to the ledger they can no longer be modified without breaking the chain of cryptographic signatures. As the ledger is entirely open, anyone can verify the entire chain of signatures at any time.

As can be imagined as observations and orbits are added to the ledger it can soon become unwieldy in size and the time taken to verify large numbers of cryptographic signatures is non-trivial. A solution to this problem was proposed in [BM91] and in Haber and Stornetta's later paper [HS97] and comprises three key advances. Firstly, rather than cryptographic signatures linking each orbit/observation with the next in the ledger, cryptographic hashes are used instead which are orders of magnitude faster to compute. Secondly, instead of adding each new entry individually, if observations and orbits are generated at approximately the same time they can be added to the ledger in groups or 'blocks' with an overall timestamp. Finally, within these blocks the orbits/obervations are linked together using a form of binary tree called a Merkle Tree [Mer80] rather than a linear chain.

3 Proof of Work

In bitcoin and other cryptocurrencies 'Proof of Work' is fundamental to rewarding those who 'process' the network, i.e perform the tasks that allows 'transactions' to be appended to the blockchain and verified. What are our transactions going to be?

In order for any distributed system to be trusted by its users, any and all aspects that require trust must be completely trusted. In reality this is never going to be the case in any network with

more than one party so the proposed system instead is designed to be an entirely trustless system that operates under the assumption that other members of the network cannot be trusted.

In order for a ledger of this type to work in a trustless environment mechanisms must be put to in place to stop so called 'Sybil Attacks' [Dou02]. Virtually all fault-tolerant distributed systems rely on a majority of nodes in the network to be acting honestly. As no limit or restriction is put on who may join or leave the network then a bad actor could trivially create any number of Sybils, nodes under the control of a single controlling node, to overcome any consensus mechanisms in the system. The idea of 'Proof of Work' is an elegant solution to this problem and was first described in [DN92] in the context of deterring spam emails. At its core, 'Proof of Work' means that any node that wants to be a part of the network must do some amount of computational work that is easy to accomplish if you are a regular user but would require massive resources to accomplish if it was to be carried out by a large amount of Sybils.

The nature of this proof of work is where the proposed system departs radically from other projects based on the idea of distributed ledgers such as Bitcoin [Nak08]. The approach used in Bitcoin and many other networks is to repeatedly change the value for a 'nonce' in a given block and then calculate the blocks cryptographic hash. If the resultant hash has certain characteristics, typically a certain number of zeros at the start, then the block can be added to the ledger with the included nonce value. Once the nonce value has been found it is very easy to verify it is true as this takes just a single iteration of a hashing function, rather than the potentially millions of iterations needed to find it.

The obvious downside to this is that an extraordinary amount of computational power is dedicated to what basically amounts to guessing numbers and then hashing them with the block. This meets the Bitcoin networks needs in terms of proof of work but at a massive cost in wasted computational power and thus energy (estimated in [Vra17] to be on the order of 100MW).

4 Useful Proof of Work

A key part of this proposal is the use of a useful proof of work algorithm based on standard processes required in the generation and maintenance of a space object catalogue. In this way the computation required to keep the network validated and thus trusted is useful to the overall aims of the network.

In general terms a proof of work algorithm requires 3 parts[BRSV17].

- Gen(1^n) is a randomized algorithm that produces a challenge c.
- Solve(c) is an algorithm that solves the challenge c, producing a solution s.
- Verify(c, s) is a (possibly randomized) algorithm that verifies the solution s to c.

While Gen and Verify should run very quickly, there should be a notion of hardness for Solve's runtime. Many of the tasks required for the generation and maintenance of a space object catalogue meet these criteria or can be modified to meet them. These include:

- Orbit Determination
- Validation of orbit predictions
- Conjunction Analysis
- Object characterisation - determining parameters from historical data
- Monte-carlo analysis - uncertainty propagation

Transactions that are computationally impractical to reverse would protect the integrity of the blockchain based catalogue. Rather than attempting to include all of the above methods in the SpocChain proof of work algorithm instead SpocChain will use orbit determination/prediction and validation of these orbits as its proof of work. The other types of proof of work could be added to the protocol at a later date as 'pegged sidechains'. This allows for simplicity in the initial protocol while giving a clear roadmap for future proof of work sidechains, of which conjucntion analysis, object characterisation and uncertainty propagation are proposed here.

5 Structure of SpOCChain

The fundamental element of SpOCChain is a transaction. We define a transaction as a cryptographically signed observation/determined orbit set. These transactions, once verified and added to the chain are finalised by allocating tokens to the contributing nodes.

One problem is that we do not know if the observation used to create the transaction is real. For this we treat each observation initially as pending until the majority of the network deems it trustworthy. This trustworthiness is ascertained by the using the observation to predict the future. If the observation (and derived orbits) are proved to be correct by subsequent observations then they are treated as canonical.

6 Conclusion

The SpOCChain project will focus on creating a functional, validated and trustworthy distributed network for the determination and dissemination of RSO tracking data and orbits using a peer-to-peer network and associated distributed ledger. The network timestamps observations and orbits by hashing them into an ongoing chain of orbit determination based proof of work, forming a record that cannot be changed without redoing the proof of work.

The network will be bootstrapped using the electro-optical and radio frequency tracking data generated by the project partners. Observations and orbits will be appended to a distributed ledger in blocks using a proof of work algorithm. A number of useful proof of work algorithms based around orbit determination will be tested.

This network, or distributed ledger, will be open and available to all for utilisation, verification and re-processing, acting as a foundational technology for truly global space traffic management.

References

[BM91] Josh Benaloh and Michael de Mare. Efficient Broadcast Time-Stamping. Technical report, 1991.

[BRSV17] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of useful work. Cryptology ePrint Archive, Report 2017/203, 2017. https://eprint.iacr.org/2017/203.

[DN92] Cynthia Dwork and Moni Naor. Pricing via Processing or Combatting Junk Mail. In *Advances in Cryptology — CRYPTO' 92*, Lecture Notes in Computer Science, pages 139–147. Springer, Berlin, Heidelberg, August 1992.

[Dou02] John R. Douceur. The Sybil Attack. In *Peer-to-Peer Systems*, Lecture Notes in Computer Science, pages 251–260. Springer, Berlin, Heidelberg, March 2002.

[HS91] Stuart Haber and W. Scott Stornetta. How to Time-Stamp a Digital Document. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology-CRYPT0' 90*, volume 537, pages 437–455. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991.

[HS97] Stuart Haber and W. Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28–35. ACM, 1997.

[Mer80] Ralph C Merkle. PROTOCOLS FOR PUBUC KEY CRYPTOSYSTEMS. page 13, 1980.

[Nak08] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. page 9, 2008.

[Vra17] Harald Vranken. Sustainability of bitcoin and blockchains. *Current Opinion in Environmental Sustainability*, 28:1 – 9, 2017. Sustainability governance.