**Problem Statement Exploration**

Users on the internet have a limited number of options when attempting to verify whether or not a file available to them has changed. Current solutions include web archives which provide a historical reference point for web sites and web resources; checking the timestamp of the webpage to which a resource belongs; manual comparisons; or relying on some other extant system.

Not all users of the internet have readily available access to a usable repository from which to check the integrity of a file and by association the trustworthiness of the underlying infrastructure (which includes the host of the resource). A Trusted Distributed Ledger (TDL) can be used to verify and ensure the integrity of a file which has a potentially untrustworthy host within a Distributed Ledger. Current users do not have access to a TDL as they can typically require large computing requirements which can be unfeasible for the average user and can also require a distributed (and geographically distributed), peer-to-peer network with a substantial number of participants. Some extant systems which attempt to solve similar problems have drawbacks. These drawbacks and the underlying technologies which shall be explored.

**Related Research, Systems and Technologies**

**HTTP extension which is more secure than HTTP but not meant to replace HTTPS**

Bayardo (Bayardo et al. 1182-1183) proposed an extension to one of the underlying protocols of the internet, Hypertext Transfer Protocol (HTTP) which they described as Merkle Tree Authentication of HTTP Responses. They wished to use existing technologies in order to create a system which increased the security of the aforementioned protocol whilst having less overhead than an extant protocol, Hypertext Transfer Protocol Secure (HTTPS) which is more secure but more expensive to implement. The underlying infrastructure which was proposed by Bayardo will be examined.

Merkle Tree Authentication of HTTP Responses would have a mirror host which would access all the resources (such as images and files) of a given site and would calculate the hash along with the URL path associated with said resource. The hash values of these files and web paths would be the leaves of the Merkle Tree of that site and would be grown. When the host server received an HTTP Get request from a client, the Host server would send the appropriate response and a Mirror Host would send the appropriate hash values. The client side web browser would calculate the hash values of the received resources and would compare the Hash value against that received from the Mirror Host. If the hash values are incongruent then an HTTP status code of “Success” would be given and the web resource would be displayed. Else the hash values are incongruent and an HTTP status code of “Not Found” (404) would be displayed and the web site would not be displayed.

The researchers proposed the use of extant, secure delivery systems for the root hash value from the Mirror Host to the Client. These systems include DNS-SEC; HTTPS to Content Provider Server and Certified PKI Signed Root Hash.

Conclusion, Strengths and Weaknesses

The implementation of this extension requires that hash values be recalculated and edited by the Mirror Hosts whenever a change is made to a website. These Mirror Hosts would thus have to download content to calculate the hash functions.

The use of an appropriate hash function and Merkle Trees provides an additional layer of authentication and protection against Man-in-the-Middle Attacks as if the Client and the Host can agree on the hash value of the sent item, it is feasibly improbable for an alteration to have been made on the content.

As the proposed system is an extension of HTTP and effectively meant to replace it. It would require the appropriate Mirror Host downloading all content from the originator sites on a frequent basis (or at least checking whether a change had occurred) to ensure the Merkle Tree of each site is up to date (this would require huge expenses and would presumably require a centralized governing body). It does not protect the client from changes made to web repositories by informing them of the change and if it did, any change made to a repository will alter all parent hash values which would be impractical to convey in its entirety.

The extended protocol proposed is theoretically workable however it requires significant, expensive infrastructure to implement. Extant web infrastructures would also need to be revised in order to use the updated protocol. Due to the scale and infrastructure requirements, semi-distributed computing would be required.

**HTTPi**

(Singh et al. 659-668) furthered the work proposed by Bayardo et al. and defined and implemented a protocol HTTPi which they theorise should perform just as well or better than HTTP in terms of security whilst providing a guarantee of the end-to-end integrity of the sent content.

The result of this research, which was carried out on a small test platform, indicate that HTTPi had worse End-to-End response times than HTTP and HTTPS whilst being less trustworthy and less secure than HTTPS.

The simulations which the researchers ran determine that the user-perceived latency is 0.7-2 seconds greater than either HTTP or HTTPS. The researchers also determine that HTTPi should replace HTTP where possible but they do not lay out a plan to achieve this.

It can be determined that the use of Merkle Trees is a feasible data structure for authenticating content online.

**Distributed peer to peer file sharing, with parallels to a RAID configuration**

Michalakis (Michalakis et al. 145-158) also proposed a distribution system similar to Bittorrent (a file sharing application which makes use of the Bittorrent Protocol) which is a decentralized peer-to-peer file sharing platform called Repeat and Compare. It aims to ensure the integrity of content between untrustworthy peers. This system is intended to store content over a network where peers do not necessarily have access to other’s content in its original form. A content creator propagates their content, which is subdivided and encrypted, across different peers and the system detects whether the content has changed through propagation. The underlying challenge which Repeat and Compare faces is the “He said, she said” problem (this sounds like a subset of the Byzantine Generals Problem). Repeat and Compare attempts to store multiple, full copies of content over the entire peer-to-peer network. It is capable of detecting and “cleansing” misbehaving nodes even when a large portion of nodes are misbehaving.

**Bitcoin: The first Peer-to-Peer, distributed ledger based cryptocurrency**

In Nakamoto’s white paper “Bitcoin A Peer-to-Peer Electronic Cash System” he presented Bitcoin which was the first peer-to-peer, decentralized cryptocurrency and it makes use of a distributed ledger in the form of a blockchain, once an element is added to this blockchain, it becomes infeasible to alter unless consensus is reached by a majority of the computational power. By making use of a peer-to-peer network and Proof-of-Work algorithms, a public, distributed ledger of transactions is kept.

**WebArchiving - Wayback Machine (Not Sourced)**

Wayback Machine is an opt-out of digital archiving system. It attempts to create a time dependent archive of a website as it is found on the internet.

Some of the limitations of the Wayback Machine are that it only has access to publicly available websites (therefore if login is required then the system does not provide an archive of it); it has a huge infrastructural footprint (it stores petabytes of data and needs to download and upload portions of this data regularly); it is not a truly comprehensive archive of the internet.

**Business Applications**

Look at what de Beer’s is doing as well as other traditional businesses which also make use of Blockchain Technologies.

**Possible Avenues of Enquiry (no sources)**

The core of the artefact I wish to create is a distributed peer-to-peer ledger which is built upon Merkle Trees and other blockchain associated technologies.

The possible applications of this particular artefact are somewhat varied and they include: a size efficient type of web-archival system where any person who wants to can add the hash of an object into the blockchain where it will be stored and used to efficiently determine whether a change has occurred to that object as experienced by the original person or anyone else, i.e. a generic peer-to-peer distributed ledger publicly available; the use of a pseudo-cryptocurrency in order to provide incentives to use the system and minimize abuse of the system; application of a blockchain to a generic business scenario; a learning tool for this type of technology.

Comments on a solely singular business oriented Peer-to-Peer Distributed Ledger. If a business is entirely responsible for their own Distributed Ledger it would require them to rent/own the hardware and other infrastructure required. If every business which could make use of Distributed Ledger Technology made use of their own infrastructure this would result in massive energy consumption (a single Bitcoin transaction requires at least 77 KWh) and excessive, and potentially wasteful, hardware costs (ASIC miners and GPU mining are notoriously expensive especially due to the rise in value of cryptocurrencies).

I would like to explore whether a general purpose solution which a lot of different entities have access to (e.g. BKB, Mercedes-Benz and the public can use) would be more energy efficient, cheaper to maintain, produce less e-waste and be more secure than a solution specific to a single entity. E.g. whether the benefits scale inversely or proportionally with size.

**Source List**

Singh, Kapil, et al. "Practical end-to-end web content integrity." Proceedings of the 21st international conference on World Wide Web. ACM, 2012.

Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).

Bayardo, Roberto J., and Jeffrey Sorensen. "Merkle tree authentication of HTTP responses." Special interest tracks and posters of the 14th international conference on World Wide Web. ACM, 2005.

Michalakis, Nikolaos, Robert Soulé, and Robert Grimm. "Ensuring content integrity for untrusted peer-to-peer content distribution networks." Proceedings of the 4th USENIX conference on Networked systems design & implementation. USENIX Association, 2007.

King, John Leslie. "Centralized versus decentralized computing: organizational considerations and management options." ACM Computing Surveys (CSUR) 15.4 (1983): 319-349.