Original

All participants in a given network cannot assume that all other participants of the network are trustworthy.

Reworked (Dr Barnard says it is better but it will need working)

Not all users of the internet have readily available access to a trustworthy Distributed Ledger to verify and ensure the integrity of a file which has a potentially untrustworthy host, in a Distributed Ledger.

All web users implicitly rely on web resources being relatively trustworthy and immutable.

* Step 2 is to do initial literature search
  + Plan what you are looking for
    - Key words; what are you looking for in title and abstract?
  + Use reputable sources
  + Do the search
  + Come back with at least three initial sources

Merkle Tree, Distributed Ledger, Peer-to-Peer Network

* Preparation for tomorrow
  + Step 3 – screen articles (need min of 3)
    - Determine if sources are relevant (e.g. does it match key words?)
      * Yes – get full article (step 4)
      * No - file (possible later use, if relevant) or discard
  + Step 5 - Start analysing and synthesising (use sticky notes; highlight; …)
    - Start drafting / organising ideas to create the background (context) of your project
    - Bring a draft written page explaining your problem to be solved and some background information (synthesised from articles you have read)
    - Will be used tomorrow

Users on the internet have a limited number of options when attempting to verify whether or not an available file has changed. Current solutions include web archives which provide a historical reference point for web sites and web resources; checking the timestamp of a resource; completely manual comparisons; relying on some theoretical or extant system.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 Trustworthy 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. The extant systems have drawbacks which shall be explored.

Bayardo et al. (Bayardo et al. 1182-1183) proposed an extension to one of the underlying protocols of the internet, Hypertext Transfer Protocol (HTTP). They wished to use existing technologies in order to create a system which increased the security of the aforementioned protocol whilst decreasing the overhead present in 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 et al. will be examined.

Their proposed system is an extension to the Hypertext Transfer Protocol which would send the hash of a web repository when an HTTP Get request is received by the server to the client, the client would then test whether what was received matches the hash value sent by the server and would either send a HTTP status code of success (if the hashes match) or an HTTP status code of Not Found (if the hashes do not match). A Merkle Tree would be grown where the leaves are the resources to be sent to the client. (Bayardo et al. 1182-1183) proposed the use of extant, secure delivery systems for the root hash value such as DNS-SEC, HTTPS to Content Provider Server, Certified PKI Signed Root Hash. The implementation of this revision requires that hash values be recalculated and edited whenever a change is made to a web repository and that mirror hosts should exist which download content to calculate the hash functions.

The solution proposed by Bayardo et al. aims to be an extension of HTTP which will replace HTTP, as such according to the guidelines provided, it would require the download of content from all originator sites on a frequent basis in to ensure the Merkle Tree’s of each site are 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 protocol proposed by Bayardo et al. 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. The revised protocol does not inform users of any alterations.

See (King1983\_CS15\_4\_CentralizedDecentralizedComputing.pdf)

See (www2005\_rj.pdf) for further details

(Singh et al. 659-668) further the work proposed by Bayardo et al. and define a protocol HTTPi which they theorise should perform just as well or better than HTTP whilst providing a guarantee of the end-end integrity of the sent content. The result of this research 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 more than either HTTP or HTTPS. The researchers also determine that HTTPi should replace HTTP where possible but they do not express a mechanism to achieve this.

Michalakis et al. proposed a distribution system similar to Bittorrent which is a decentralized peer-to-peer file sharing platform.

(Michalakis et al. 145-158) proposed a system called Repeat and Compare which is a Peer-to-Peer Content Distribution Network. 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 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.

Possible Sources

**Practical end-to-end web content integrity**

https://dl.acm.org/citation.cfm?id=2187926

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

# Merkle tree authentication of HTTP responses

https://dl.acm.org/citation.cfm?id=1062929

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.

(10.1.1.79.9839.pdf)

**SiaCoin**

https://siawiki.tech

**Ensuring content integrity for untrusted peer-to-peer content distribution networks**

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.

https://dl.acm.org/citation.cfm?id=1973441

see Michalakis.pdf

# Centralized versus decentralized computing: organizational considerations and management options

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

Immutable, trusted source

Source of problem

Jumped into sources which attempt to solve similar/related problems

Explore how HTTPi is a failure and other failings of the research.

Laymen Terms -> indepth look -> laymen successes and failures -> indepth/more focused look

<https://archive.org/help/wayback_api.php>

Mine Wayback api?