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.

* 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

(Bayardo et al. 1182-1183) proposed 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 proposed solution 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 order 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.

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.

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