

Property of a good Hash Function.

* Block 1: One way hash Function.

→ Map variable length i/p string to fixed length string.

* Easy to hash i.e. easy to get $\text{Hash}(x)$ for x .

* For any given $\text{Hash}(x)$ it is almost impossible to get x .

* Almost impossible to have two files say x & y having same hash.

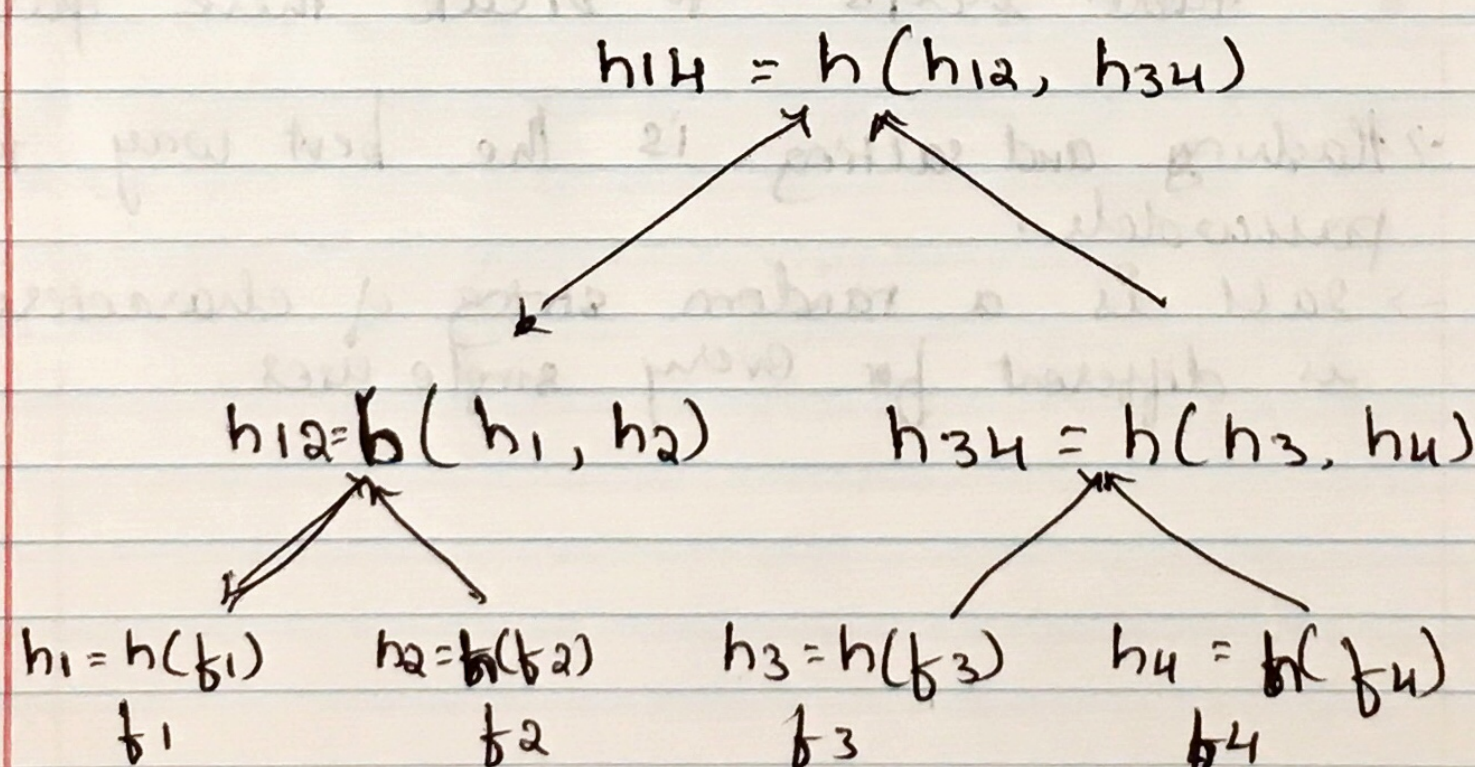
* Minor change in x , large change in $\text{Hash}(x)$.

Merkle's Hash Tree

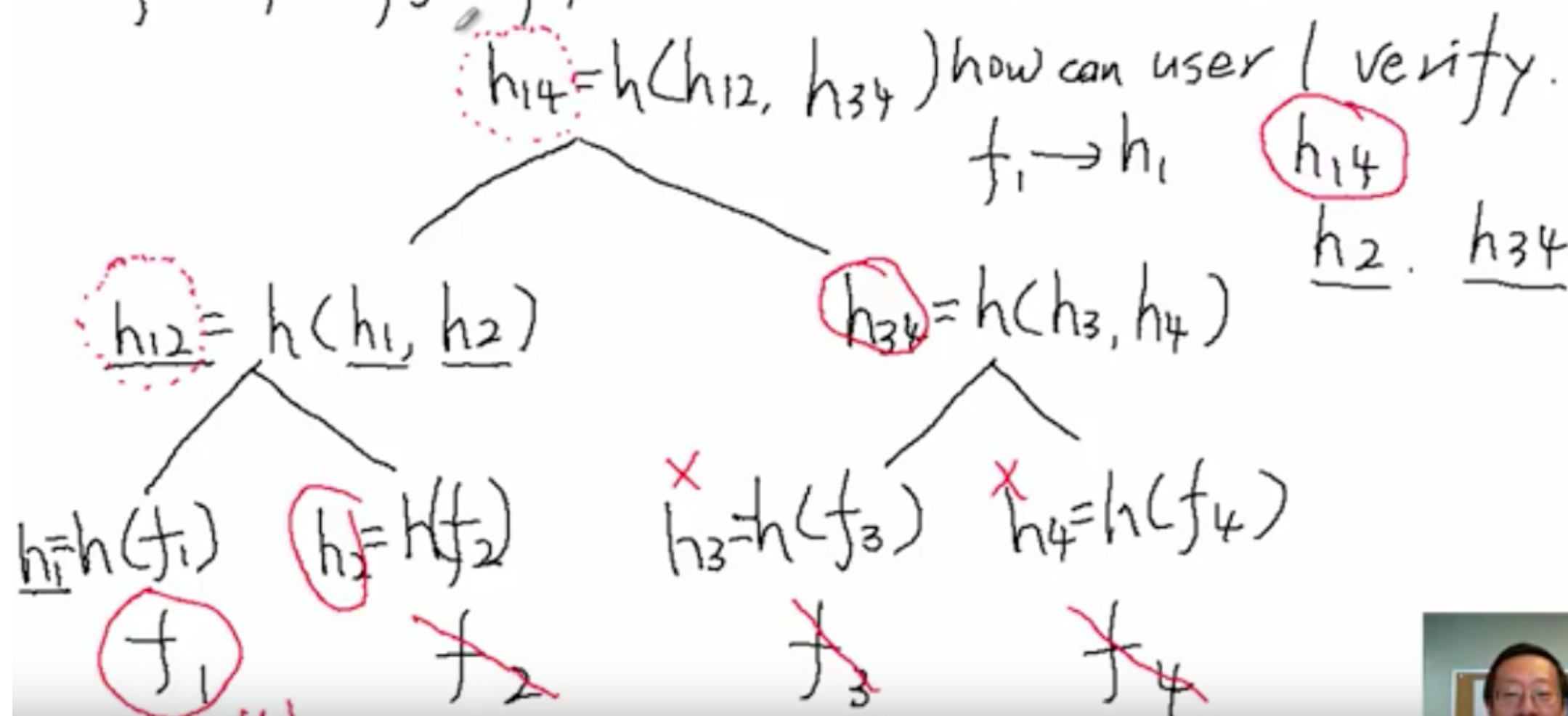
3.7 Merkle's hash tree

→ Binary tree, every time number of nodes reduced by half.

Say we have 4 files f_1, f_2, f_3, f_4 , then tree



f_1, f_2, f_3, f_4 tree. publish h_{14} :



- The beauty of the Merkle's hash tree is even the the user has h_1 , and provided with h_2, h_{34} , the file f_2, f_3 and f_4 can't be found by hashes.

Real time example of Merkle's hash tree is to use in checking the integrity of online movie streaming sites, where they divide the whole file into f_1, f_2, \dots, f_{100} .

movie online.

(4G) integrity movie.

Digitally signed.

Signature $(h_{1,1000})$

many clips: $f_1, f_2, \dots, f_{1000}$

