Hash Calendar

DS Presentation

Prof. M.M. Gore

Motilal Nehru National Institute of Technology Allahabad
Computer Science & Engineering

May 3, 2023

Team Members

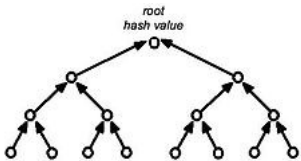
Name	Registration No.
1. Aman Jain	2022CA009
2. Ayush Vardhan	2022CA018
3. Devyani Pathak	2022CA028
4. Dharamendra Singh	2022CA029
5. Manas Kumar Giri	2022CA050
6. Prabhat Tripathi	2022CA060
7. Rishabh Verma	2022CA080
8. Sonika Panth	2022CA103

Table Of Content

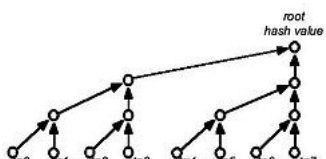
Introduction and Representation

Introduction

- A Hash calendar is a special type of Merkle or Hash tree.
- It was invented by Estonian cryptographers Ahto Buldas and Mart Saarepera.
- A Hash calendar is data structure that is used to measure the passage of time by adding hash value to an append only database with one hash value per second.
- It comes with the property that at any given moment the tree contains a leaf node for each second.



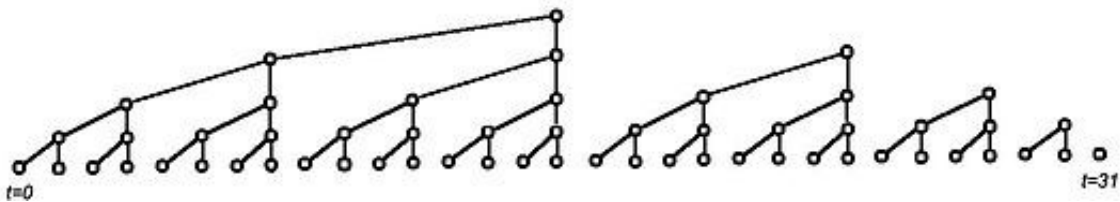
Hash Tree



Hash Calendar

Representation in Terms of Seconds and Leaf Node

Here is a hash calendar after 31 seconds...



The hash chains can be extracted from any hash trees. Hash calendars are built in a very deterministic manner such that the shape of a tree for any moment can be reconstructed by knowing just the number of leaf nodes in the tree.

Construction

Construction

There are different algorithms that can be used to construct the hash calendar and extract a relevant hash chain per second.
In easiest way we can imagine the construction of hash calendar in two phases...

1. First phase: Here the leaves are collected into complete binary trees, starting from left, and making each tree as large as possible.

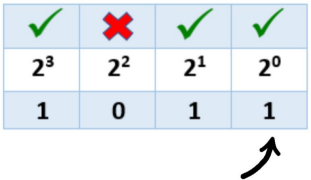
2.Second phase: In the second phase, the multiple unconnected trees are turned into a single tree by merging the roots of the initial trees, but this time starting from the right and adding new parent nodes as needed (red nodes).

Let's Take an Example:

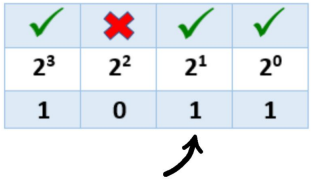
Let's create a hash calendar for 11 seconds
First of all we will convert 11 into its binary form
$(11)_{10} = (1011)_2$
when there is 1 we create a hash tree for that with 2^n *nodes*
where n is the position of that bit from right starting from zero.

$$(11)_{10} = (1\ 0\ 1\ 1)_2$$

This is the hash tree of single node:



Now, we move to next bit



let's see how we add two single nodes to create a hash tree of two leaf nodes

✓	✗	✓	✓
2^3	2^2	2^1	2^0
1	0	1	1

Here we will not create hash tree as it is 0 bit

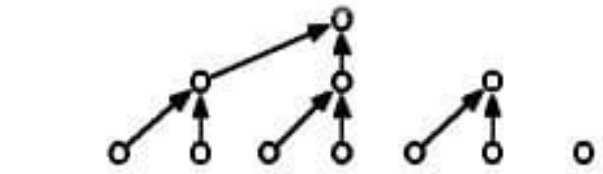
✓	✗	✓	✓
2^3	2^2	2^1	2^0
1	0	1	1

let's add two double node structures to create a hash tree of 4 leaf nodes

✓	✗	✓	✓
2^3	2^2	2^1	2^0
1	0	1	1

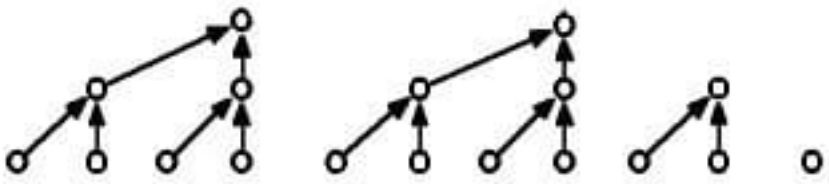
let's add two double node structures to create a hash tree of 4 leaf nodes

✓	✗	✓	✓
2^3	2^2	2^1	2^0
1	0	1	1



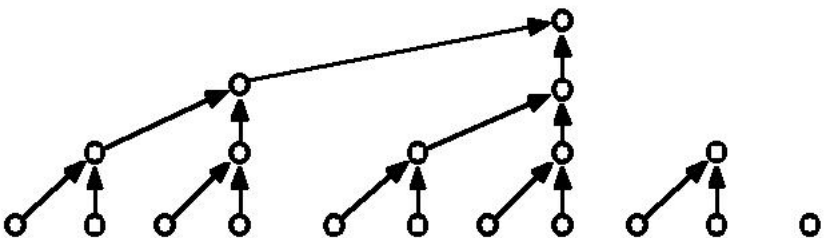
let's add two 4 nodes structures to create a hash tree of 8 leaf nodes

✓	✗	✓	✓
2^3	2^2	2^1	2^0
1	0	1	1

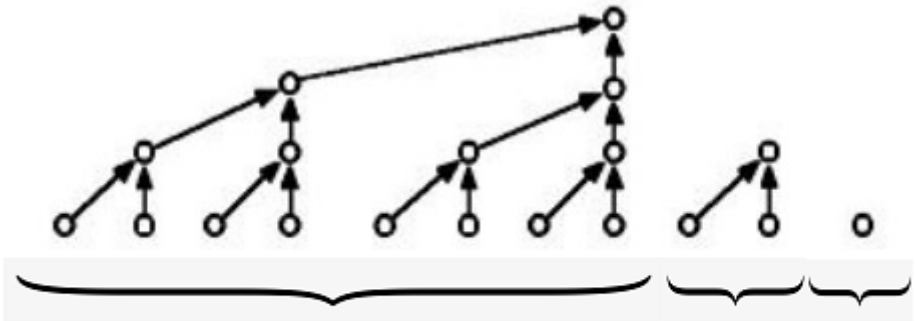


let’s connect two 4 nodes structures to create a hash tree of 8 leaf nodes

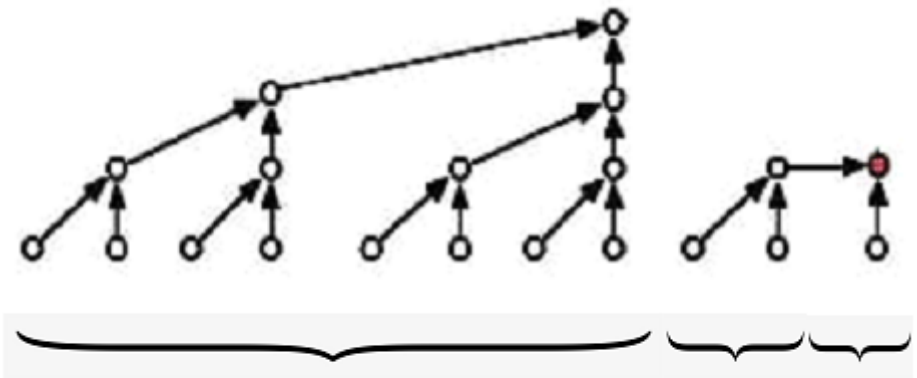
✓	✗	✓	✓
2^3	2^2	2^1	2^0
1	0	1	1

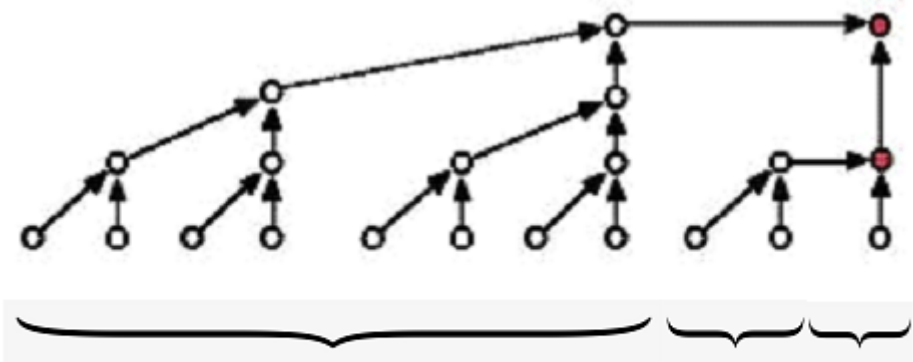


Now add those two 4 leaf node structures



second phase started





$2^3 = 8$ $2^1 = 2$ $2^0 = 1$

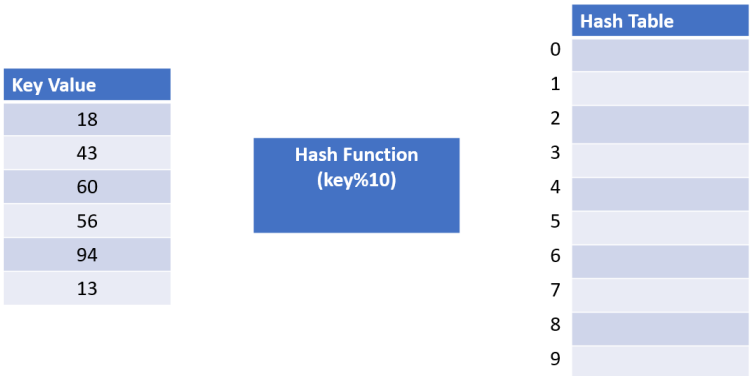
Hashing

Hashing

- Hashing is the process of transforming any given key or a string of characters into another value.
- In hashing,
  - Larger keys are converted into smaller keys by **hash function**
  - The value then stored in data structure called **hash table**
  - When more than one value to be hashed by a hash function to the same slot in the hash table then it is called **collision**
  - When a collision occurs, the new key-value pair is added to the linked list at the corresponding index, this process is called **hash chaining**.

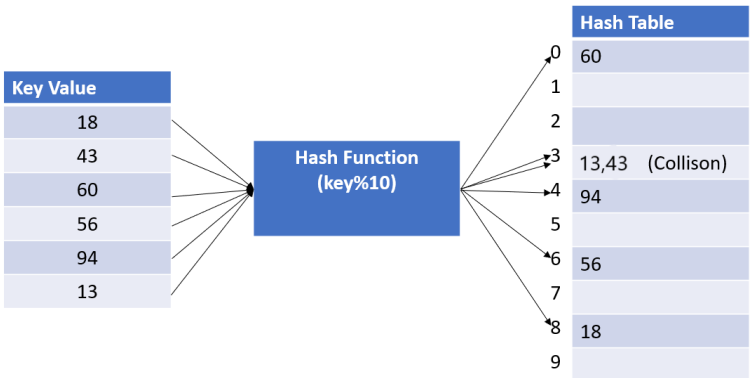
Let's Take an Example

Here, we have some keys, a hash function and a hash table.

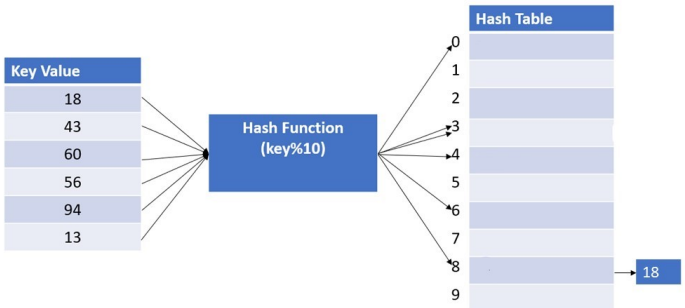


Let’s Take an Example

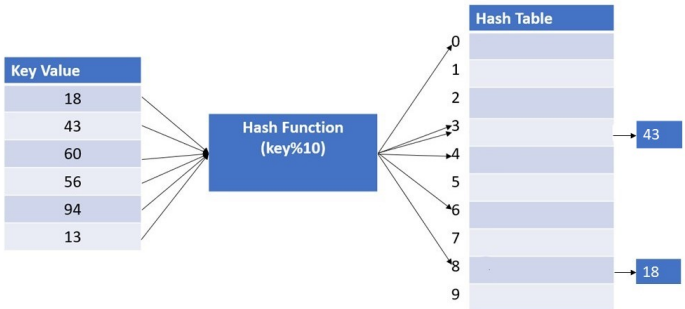
Here two key values are at same index so collison occurs.



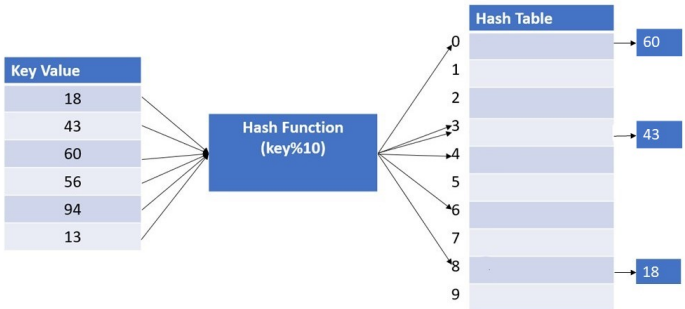
Let’s Take an Example of Chaining



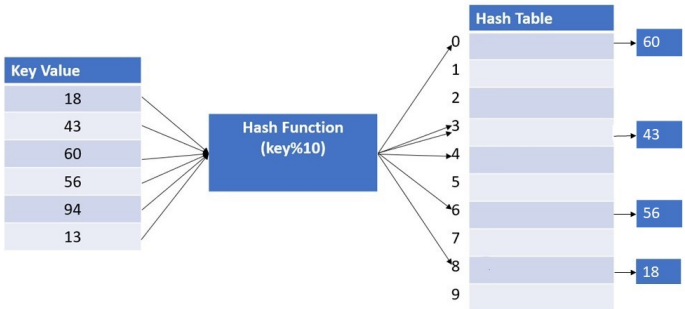
Let’s Take an Example of Chaining



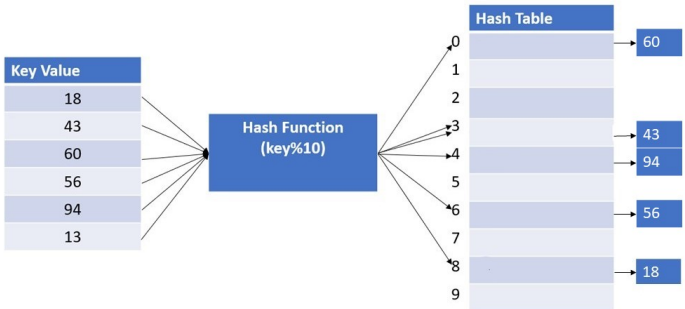
Let’s Take an Example of Chaining



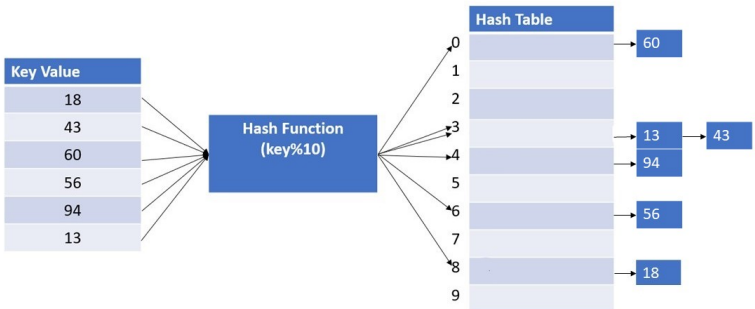
Let’s Take an Example of Chaining



Let’s Take an Example of Chaining



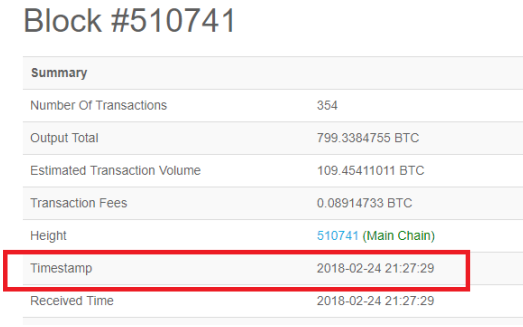
Let’s Take an Example of Chaining



Linked Timestamping
&
Hash Calendar Applications

Linked Timestamping

- Proposed by Stuart Haber and W. Scott Stornetta in 1990.
- Linked timestamping is a process of adding a digital timestamp to a document or data file to verify its authenticity and integrity.
- Linked timestamping is often used in legal and financial contexts, where it is important to have a verifiable record of a document or when a transaction is occurred.



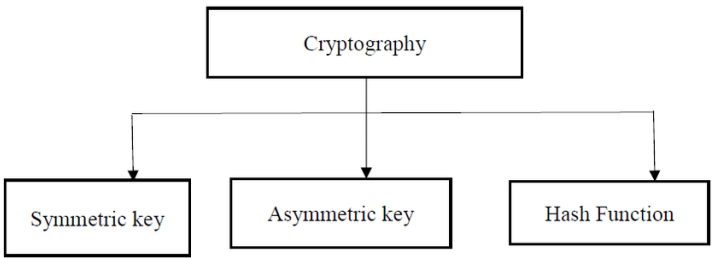
Applications of Hash Calendar

- **Digital forensics**: Hash calendars can be used to analyze digital data based on date and time, such as email communications or file modifications. This can help investigators identify patterns or timelines related to a particular case.
- **Scheduling software:** A hash calendar can be used to manage appointments, meetings, and other events in scheduling software.
- **Time-series databases:** A hash calendar can be used to index the data by date or time period, making it easy to retrieve data for a specific time period.
- **Event logging:** Hash calendars can be used to log events, such as server errors or user actions, based on date and time.
- **Online advertising:** Hash calendars can be used to manage online advertising campaigns that are scheduled to run for a specific time period.

Cryptography and Hash Function

Cryptography

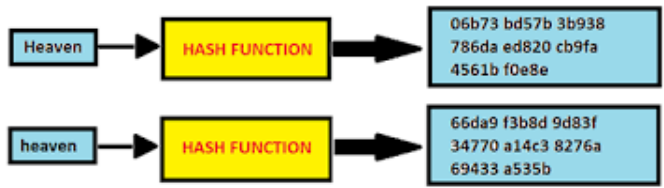
- Cryptography is a combination of two words ”crypt” means hidden and ”graphy” means writing.
- Cryptography is a technique of securing information and communications through the use of codes so that only those person for whom the information is intended can understand it and process it.
- Cryptography remains important for protecting data and users, and preventing cyber criminals from stealing the data.



Cryptographic Hash Function

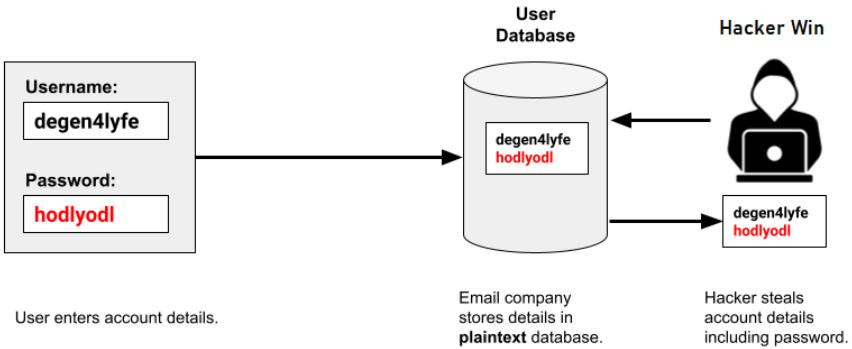
Hash Functions:

1. Cryptographic Hash Function is a hash function that takes random size input and yields a fixed-size output.
2. It is easy to calculate but challenging to retrieve the original data.
3. It is strong and difficult to duplicate the same hash with unique inputs and is a one-way function so revert is not possible.



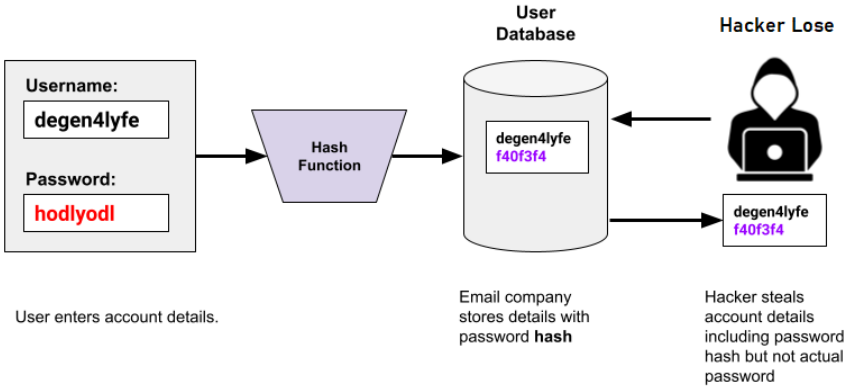
Example of Hash Function: Online Passwords

When you create an email address and password, your email provider runs the password through a hash function and saves the hash of your password. But Why??



Example Of Hash Function: Online Passwords

When you create an email address and password, your email provider runs the password through a hash function and saves the hash of your password. But Why??



Merkle Tree

Merkle Tree

- Merkle Tree is a hash based data structure in which each leaf node is a hash of block of data and each non-leaf node is a hash of its children.
- The concept of Merkle Tree was named after Ralph Merkle in 1979.
- Hash Trees or Merkle Trees are used in hash-based cryptography.
- Hash Trees can be used to verify any kind of data stored, handled and transferred in and between computers.
- In short, a Merkle Tree formation is the process of making a single hash from a group of hashes.

Illustration of Merkle Tree

H1,H2,H3,H4 are hash value of T1,T2,T3,T4.

Illustration of Merkle Tree

H1,H2,H3,H4 are hash value of T1,T2,T3,T4.

Illustration of Merkle Tree

H1,H2,H3,H4 are hash value of T1,T2,T3,T4.

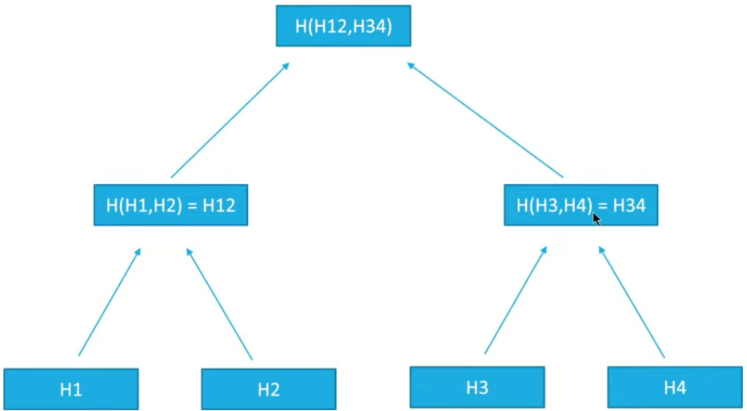


Illustration of Merkle Tree

H1,H2,H3,H4 are hash value of T1,T2,T3,T4.

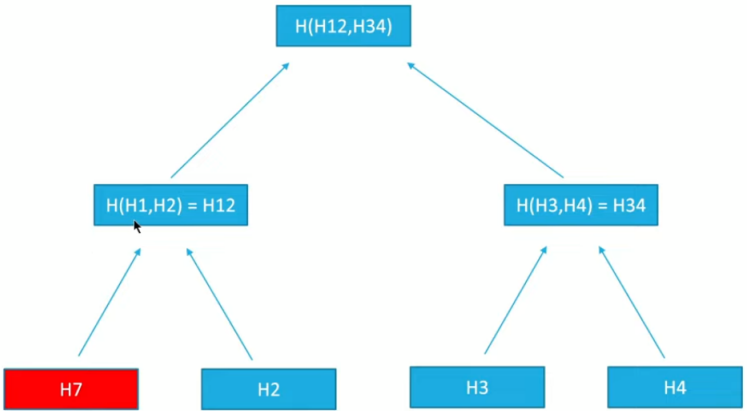


Illustration of Merkle Tree

H1,H2,H3,H4 are hash value of T1,T2,T3,T4.

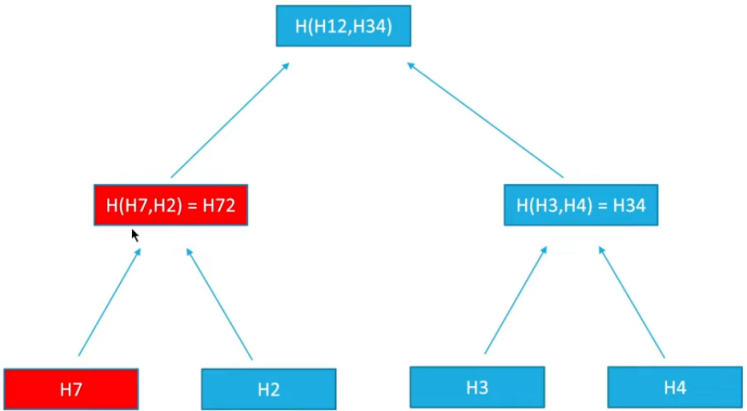
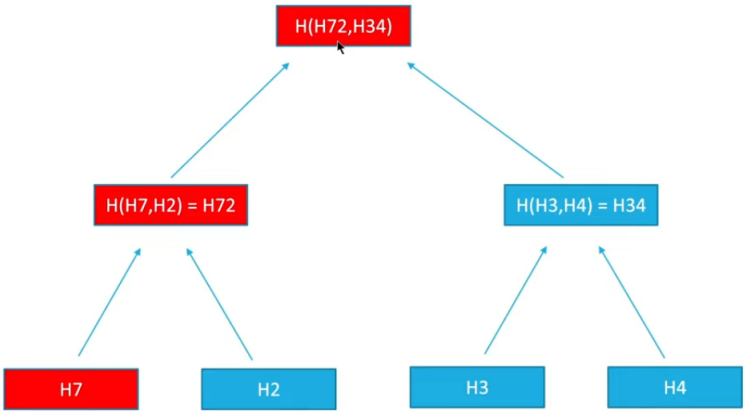


Illustration of Merkle Tree

H1,H2,H3,H4 are hash value of T1,T2,T3,T4.

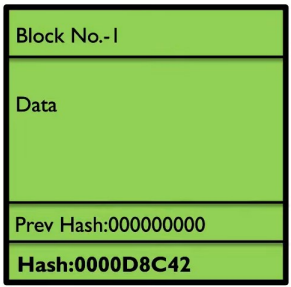


Blockchain

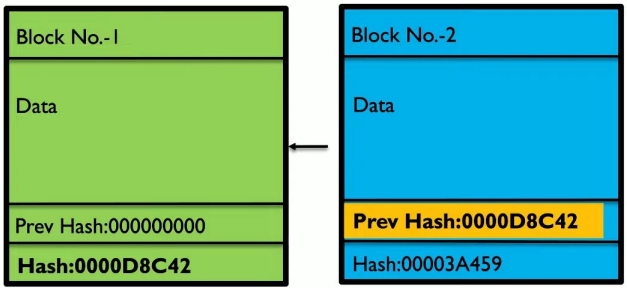
Blockchain

- Blockchain is just like database but is structurally and functionally different. While database store data using table form, blockchains store data in blocks.
- A Blockchain database is a type of digital ledger that is distributed across a network of computers. Each block contains a record of multiple transactions. Once a block is added to the chain, it cannot be altered or deleted.
- Blockchain technology has ability to provide secure, transparent and tamper-proof data storage and transfer.

Blockchain

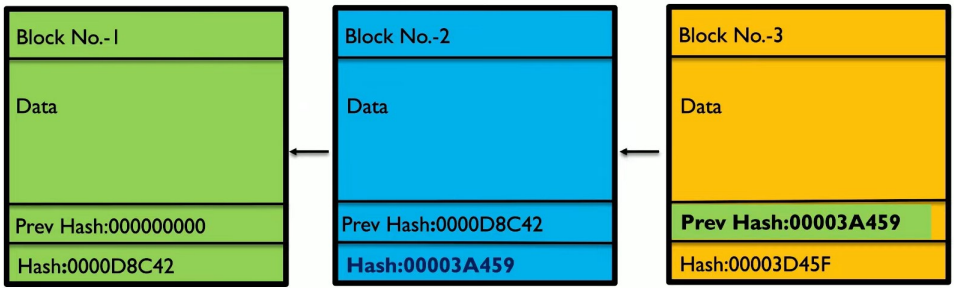


Blockchain



Genesis Block

Blockchain



Genesis Block

Use of Hash in Blockchain

- A hashing algorithm takes an infinite number of bits, performs calculation on them and outputs a fixed number of bits.
- As a result, the original data is called input and the final transformation is called a hash.
- Pointers and linked lists are used in hashing to navigate and connect the blocks.
- Each block header contains the previous block's hash, which ensures that nothing has been tampered with as new blocks are added.
- Blockchains use hashes to secure information and make the ledger immutable.

Advantages & Disadvantages of Hash Calendar

Advantages Of Hash Calendar

Data Integrity: Hash Calendar provide data integrity. By computing the hash values for each block of data and storing them in the hash calendar, any changes to the data will result in a different hash value, which can be detected during verification.
Data Backup: Hash Calendar can be used in backup and archive systems to ensure that the data has not been corrupted or lost during the backup process. By calculating the hash values before and after the backup process, any changes or corruption can be detected.
Tamper-evident: Hash Calendar can provide evidence of tampering or unauthorized access to the data. Any changes to the data will result in a different hash value, which can be used to detect and investigate any suspicious activity.

Disadvantages of Hash Calendar

Maintenance: To ensure the integrity and authenticity of the data, hash calendar require regular maintenance, including verifying the existing data and updating the calendar with any new data. Failure to maintain the hash calendar can compromise its effectiveness.
Collision Risk: Hash functions are not perfect, there is a risk of collisions, where two different pieces of data produce the same hash value. While this is rare, it can happen and it can compromise the integrity and authenticity of the data.
Size Limitations: Hash functions produce fixed-size hash values, which can limit the size of the data that can be stored in the hash calendar. While this can be addressed by dividing the data into smaller blocks, it can still be a limitation in certain situations.

Conclusion

- In conclusion, hash calendars are a useful tool for organizing and tracking data. They allow for quick retrieval of information based on a specific key, and can be implemented in a variety of programming languages.
- It is important to note that while hash calendars offer many benefits, they also come with some limitations.

Overall, hash calendars are a powerful tool that can enhance the performance and functionality of software applications.

Resources

1. Wikipedia (https://en.wikipedia.org/wiki/Hash$_c$*alendar*)
2. Educational Series on Hash Functions (http://www.guardtime.com/educational-series-on-hash-functions)
3. Merkle Tree (https:/merkle-tree-blockchain)
4. Cryptography (https://www.geeksforgeeks.org/cryptography-and-its-types/)
5. Cryptographic Hash functions (https://en.wikipedia.org/wiki/Cryptographic$_h$*ash*$_f$*unction*)
6. Blockchain (https://www.investopedia.com/terms/b/blockchain.asp)

Thank You