**Sarthak Singhal**
**(20171091)**

# PQR-3

**10th April 2020**

## Question

Recall that a pointer to data X stores the address of X, denoted by &X. Similarly, let a hash-pointer to data X be the address of X along with the cryptographic hash (say, H) of X, denoted <&X, H(X)> . Further, let a hash & sign-pointer to data X be the address of X, along with the cryptographic hash (say, H) of X that is digitally signed by the owner of X, denoted <&X, H(X), σ>. Let D be your favourite data structure among stacks, queues, lists, trees, forests, graphs, priority-queues along with their variants (like Binomial heaps, skip-lists, red-black-tress and so on). Consider a pointer-based implementation of D. It is straight-forward to visualize the corresponding hash-pointer-based implementation of D, wherein every pointer &X is replaced by <&X, H(X)>.

1. What do you think are the advantages of the hash-pointer-based implementation of D over the pointer-based implementation of D?
2. Specifically, can you think of one application/problem/setting/protocol, say $A_{hash}$ , wherein a hash-pointer-based implementation of D is more suitable?
3. Analogously, list the advantages of the hash & sign-pointer-based implementation of D. Give an application $A_{sign}$ where it is (more) suitable.

Justify your answers.

# Answer

For this evaluation, I have used stack data structure and I'm going to explain how hash-pointer based and hash & sign pointer based stacks are useful by taking examples from blockchain, cryptocurrency and bitcoin.

**Ans 1)**

A hash pointer is basically a pointer to the place where some information is stored. Together with the pointer we are going to store a cryptographic hash of the information. So a regular pointer gives us a way to retrieve the information. A hash pointer will allow us to get the information back and verify that if the information has changed or not. So a hash pointer tells us where something is and what it's value was when we last saw it. So this is the advantage of hash-pointer based implementation of D over the pointer based implementation as it helps us to tell whether some information was modified or not.

**Ans 2)**

One application where hash-pointer based implementation of D is more suitable is blockchain which is basically a stack/linked list with hash pointers. Since this is used in cryptocurrencies, security is of high importance and we don't want anyone to mess around with the stored values as it could result in huge monetary losses. Thus to prevent tampering of block information, we store a hash pointer to the previous block. If someone modifies the block contents, the contents would not match with the corresponding hash in the next block. Thus to modify the chain, the adversary would need to modify all the blocks till the end which would be computationally expensive. This could be detected easily as one remembers the head pointer which the adversary can't modify.

**Ans 3)**

Now suppose A and B are two parties recording transactions in a blockchain. With hash pointer based implementation, we can tell whether some content is modified or not. Now suppose A records a same transaction multiple times or it may record a wrong transaction that B would like to reject. This can be achieved by B signing all the legal transactions and thus he would be able to reject the faulty transactions where the message and signature do not match. This is used in cryptocurrencies like bitcoin where digital signatures help authenticate the transaction. Besides as seen above , the adversary may try to modify all blocks to allow entry of one fraudulent block. To prevent this we store the signature of the head block and then we can tell whether the head block has been modified or not by simply storing its digital signature. Thus hash & sign pointer based implementation also helps to authenticate the values. So this is the advantage of hash & sign pointer based implementation of D over the hash based implementation.

**Cryptocurrency/Bitcoin**

- Cryptocurrency is basically a decentralised ledger where each block in the blockchain contains several transactions.
- Since any one central party can’t be trusted, we have it in a decentralised way.
- If certain rules are followed, we can totally replace use of physical money with digital money.
- This leads to digital currencies like bitcoin which later may be exchanged for physical money.
- To ensure only legal transactions, parties use digital signatures to prevent addition of faulty blocks.
- To identify which transaction is illegal, we may need logs of previous transactions and thus it is necessary for logs to be synced and not to be tampered.

- Since we store a hash of the previous block, it takes a lot of computation to tamper a block in the blockchain and thus is infeasible.
- Also if one party tries to broadcast different messages to different people, it needs to have tremendous computation power which would overpower all the other parties.
- This is due to the fact we are using hash & sign based pointers which are made from one way functions and cannot be forged.