Evaluation 3

20171076

Q. A pointer to data X : $<\&X>$ where $\&X$ is the address of X

A hash pointer to data X: $<\&X, H(x)>$ where H is a cryptographic hash function

A hash & sign pointer to data X: $<\&X, H(X), \sigma>$ where σ is the digital sign by owner of X.

D is an implementation of a data structure.

a) What are advantages of hash-pointer based implementation of D over a regular pointer based implementation?

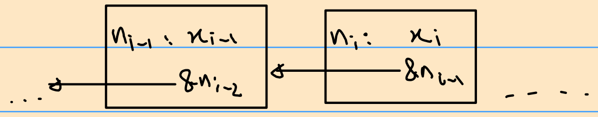
Specifically, think of one application/setting/protocol A_{hash} where a hash-pointer implementation is more suitable.

b) Analogously, advantages of hash-and-sign, and an application A_{sign}

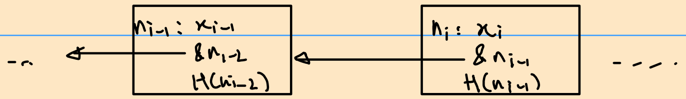
A. Consider a reversed linked-list data structure, where each node points to the previous node. It can be used for any form of sequential data storage, like in ledgers.

CONSTRUCTION OF DATA STRUCTURES

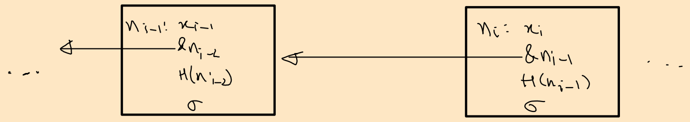
Standard pointer implementation: Each node n_i stores data x_i, and a link to the previous node n_{i-1}



Hash pointer implementation: Each node n_i stores data x_i, and (i) a link to previous node (ii) a hash of the previous node.



Hash and Sign pointer implementation: Each node n_i stores data x_i, and (i) a link to previous node (ii) A hash of the previous node (iii) a digital signature of the data, link, and hash by the data owner.



a) Advantages of hash pointer over regular pointer:

As each pointer also contains the hash of the previous block's data, if the data is modified by an adversary, they will have to either

i) Recompute the hash and replace links for every following linked node, or

ii) Replace the data in such a way that the hash remains unchanged.

Considering a PPTM Adversary with negl() error, option (ii) is not viable under the assumption that OWFs exist. But if the hash function H() is available to them, then option (i) is trivial unless external observers are monitoring the linked list for changes,

We can construct a specific protocol Ahash to fully utilise hash pointers. Create a linked list with hash pointers and share it among multiple users. Then encourage users to maintain the state of the list (ensure they constantly check for hash correctness) by giving rewards for doing so, and penalties for doing otherwise.

This is similar to blockchains (with distributed trust) that would not be possible without hash pointer.

b) Advantages of hash and sign pointer

Similar to a hash pointer, a hash and sign pointer attempts to ensure immutable data structure creation, with the added layer of a signature for secerity.

A PPTM Adversary attempting to change any data has to ensure that the hash is unchanged, or must replicate the digital signature, which is only possible with negl() probability.

Asign: We can use this data structure for a centralised ledger that is written to by a Trusted Authority (like a bank), with external verifiers confirming correctness of the hashes and validity of the signature.