Lecture #1: Intro to Crypto and Cryptocurrencies

Lecture 1.1 Cryptographic Hash Functions

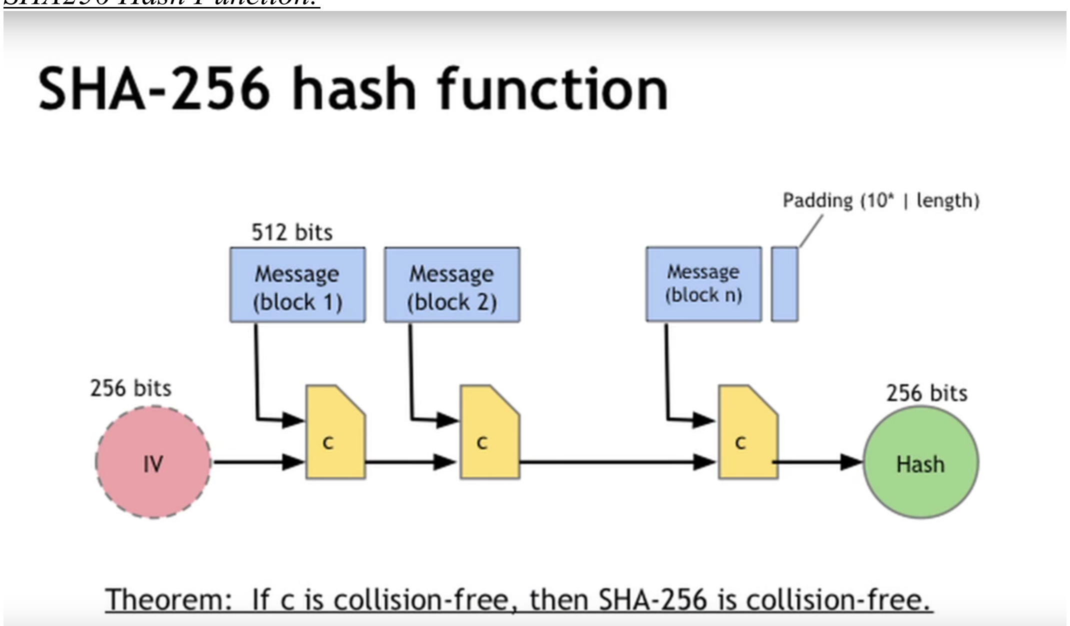
Hash Function:

Take any string as input
Fixed-size output (we'll use 256 bits)
Efficiently computable

Security properties

Collision-free: e.g x!=y and Hash(x) == Hash(y)
Hiding: Given H(x), it is infeasible to find x
Puzzle-Friendly
For every possible output value y, if k is chosen from a distribution with high min-entropy, then it is infeasible to find x such that H(k | x) = y.

SHA256 Hash Function:



Lecture 1.2 Hash Pointers and Data Structures

Hash Pointers

Pointers to where some info is stored, and
(Cryptographic) hash of the info

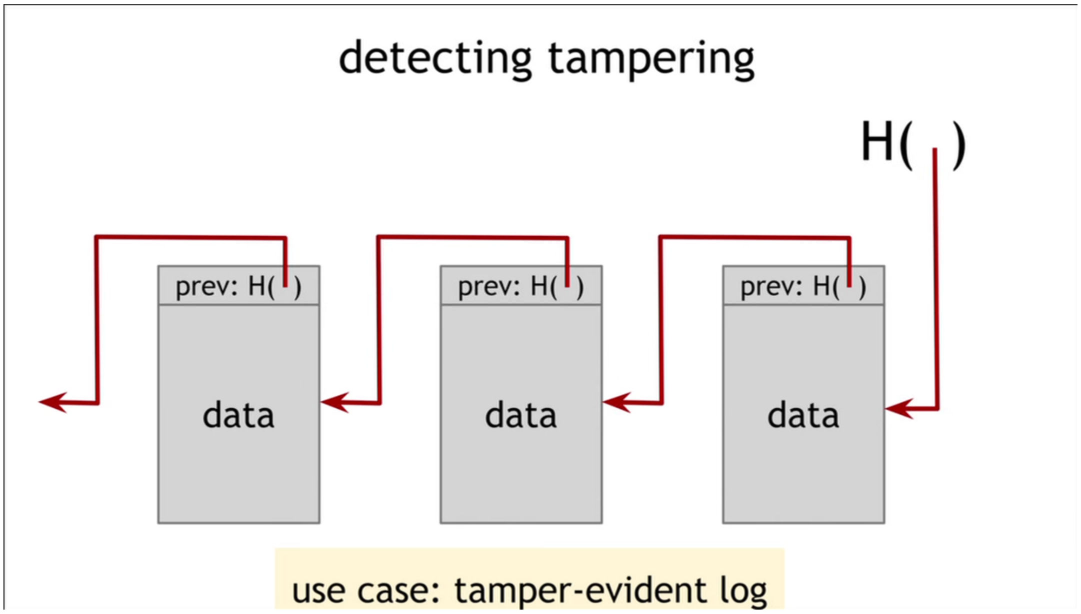
If we have a hash pointer, we can:

Ask to get the info back, and
Verify that it hasn't changed

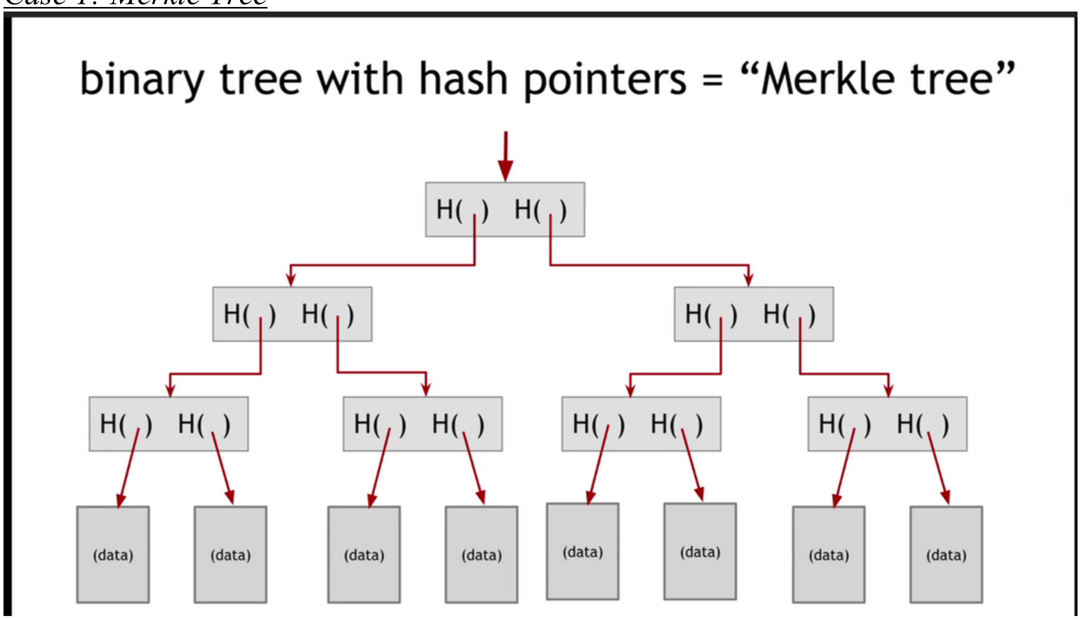
KEY IDEA

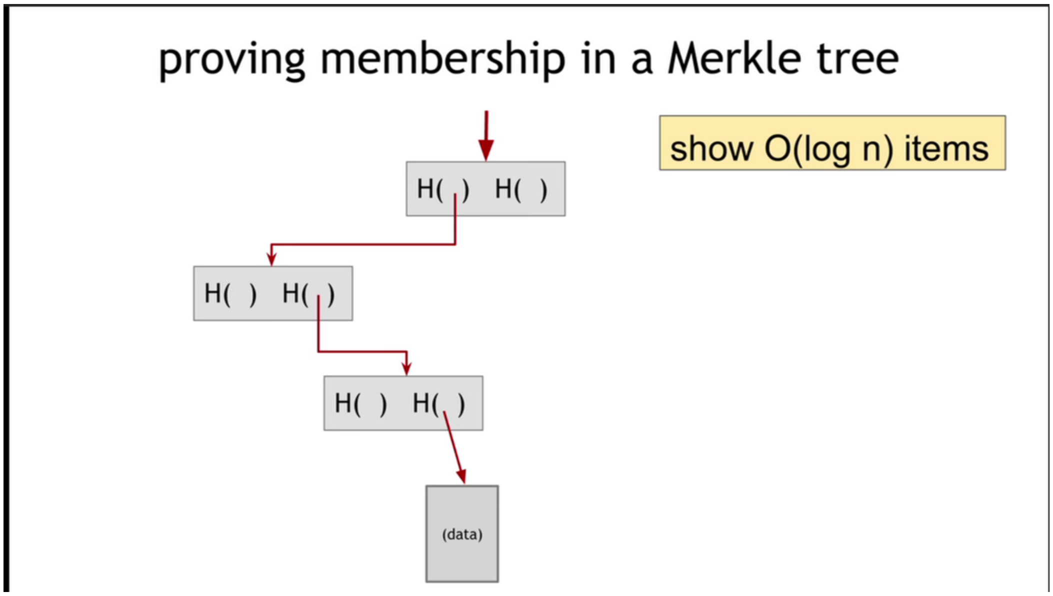
Build data structures with hash pointers

Case 1: Blockchain



<u>Case 1: Merkle Tree</u>





Advantages of Merkle Trees:
Tree holds many items, but, just need to remember the root hash
Can verify the membership in O(logN) time/space

Variant: Sorted Merkle Tree
Can verify non-membership in O(logN)

More generally..
Can use hash pointers in any pointer-based data structure that has no cycles

Lecture 1.3 Digital Signatures
Properties:
1, Only you can sign, but anyone can verify
2, Signature is tried to a paticular document, can't be cut-and-pasted to another doc

APIs:
(sk, pk) := generateKeys(keysize)
    sk: secret signing key
    pk: public verification key
sig := sign(sk, message)
isValid := verify(pk, message, sig)

"Valid signatures verify"
verify(pk, message, sign(sk, message)) == True
"Can't forge signature"

Background knowledge:
Bitcoin uses **ECDSA** standard
    Elliptic Curve Digital Signature Algorithm
Relies on hairy math

Good randomness is essential

<u>Lecture 1.4 Public Key as Identities</u>

Useful trick: public key == an identity

<u>How to create a new identity?</u>

1, Create a new, random key-pair (sk, pk)

Pk is the public 'name' you can use

[usually better to use Hash(pk)]

Sk lets you 'speak for' the identities

2, You control the identity, because only you know sk

if pk 'looks random', nobody needs to know who you are.

Privacy

Address not directly connected to real-world identity.

But observer can link together an address's activity over time, make inference

<u>Lecture 1.5 A Simple Cryptocurrency</u>

<u>Example: GoofyCoin</u>

Duble-spending attack: the main design challenge in digital currency.

<u>Example: ScroogeCoin</u>

1, Create coin transaction

2, Pay coin transaction

Consume coins valid

Not already consumed

Total value out = total value in, and

Signed by owners of all consumed coins

Critical Issue: centralization

Immutable coins

Coins can't be transferred, subdivided, or combined.

But: you can get the same effect by using transactions to

Subdivided: create new trans

Consume your coin

Pay out two new coins to yourself

Lecture #2: How Bitcoin Achieves Decentralization

Lecture 2.1 Centralization vs. Decentralization

Competing paradigms that underline many digital technologies

Aspects of decentralization in Bitcoin

Peer-to-peer network:

Open to anyone, low barrier to entry

Mining:

Open to anyone, but inevitable concentration of power

Often seen as undesirable

Updates to software:

Core developers trusted community, have great power