Danny Diekroeger @dannydiekroeger
9 Jul 20 • 30 tweets • dannydiekroeger/status/1281100866871648256

Tr

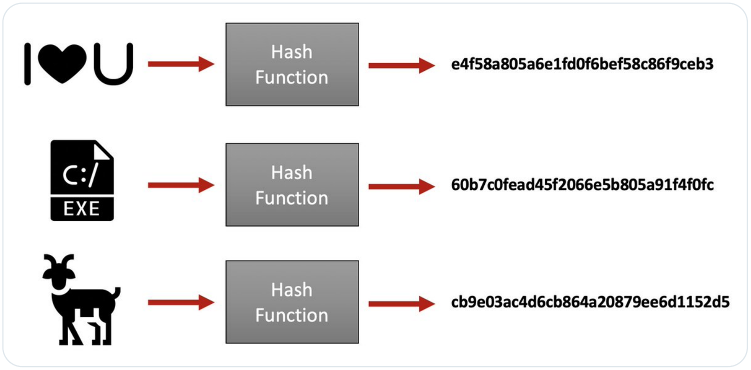
Hash Functions are a key concept in computer science, cryptography, and yes #bitcoin.

It's impossible to explain how bitcoin works without first understanding what a Hash is.

If you'd like to expand your background knowledge, read on 👇

Hashes are used for all kinds of things, but a simple analogy is to think of them like fingerprints for data.

You can think of a Hash Function like a magic fingerprint reader. You can enter any data into the hash function, and it will spit out a fingerprint.


I❤U
Hash Function
e4f58a805a6e1fd0f6bef58c86f9ceb3
C:/ EXE
Hash Function
60b7c0fead45f2066e5b805a91f4f0fc
Hash Function
cb9e03ac4d6cb864a20879ee6d1152d5

Anything can be used as a Hash Function, but a good hash function will ensure that different input data will each have unique fingerprints.

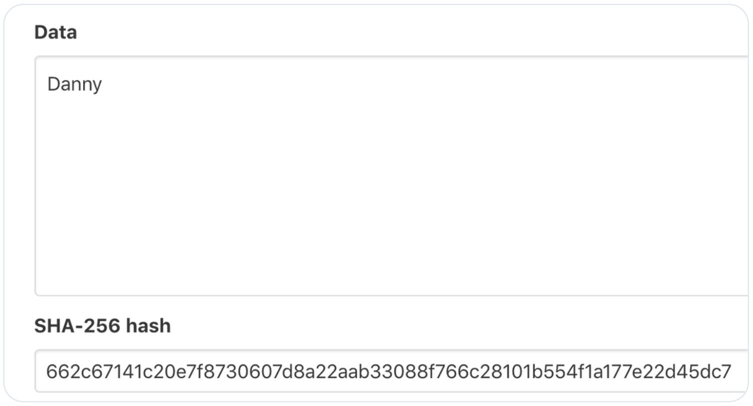
One of the most common Hash Functions used today is called SHA-256, which stands for "Secure Hash Algorithm, 256 bits"

Check out the link below. Try entering your name and see what the Hash Function spits out

Notice that no matter how much or little data you enter, the resulting hash is always different

https://xorbin.com/tools/sha256-hash-calculator

Here's what happens when I type in my name. Notice that if you type in "Danny" as well, you'll get the exact same result. This is the SHA-256 hash (or "fingerprint") of my name.



Data

Danny

SHA-256 hash

662c67141c20e7f8730607d8a22aab33088f766c28101b554f1a177e22d45dc7

But what is this SHA-256 voodoo anyway? How is it calculating that fingerprint?

Basically, it does some fancy math on the input data.

If you're curious, you can look at the details here:

https://en.wikipedia.org/wiki/SHA-2#Pseudocode

This fancy math guarantees a couple things:

1) Every fingerprint is very likely to be unique

2) If I give you just a fingerprint, you have no way of guessing what the input data was

This second point actually begins to erode the "fingerprint" analogy, but its very important.

To explain further, here's an example question...

What is the input data who's SHA-256 hash is:

04a474a12bad5d61e9d461945e6785e28b8789c591efc33a75621a2e65183393

Any guesses?

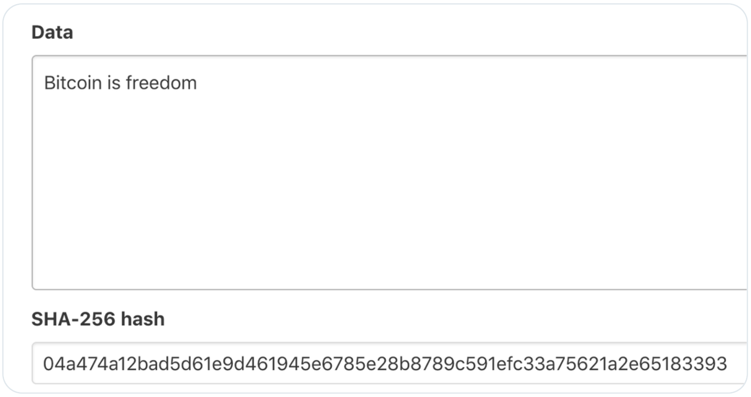
Didn't think so! According to what we know about math, it's impossible to figure out the answer to this question without just brute force guessing.

So go ahead, start guessing! The input data could be anything. Maybe its the letter "A", or maybe its the entire text from the 3rd Harry Potter book.

You'll never know unless you guess... or if I tell you...

The answer is "Bitcoin is freedom"

Go ahead and run that through the hash function to verify I'm telling the truth!



Data

Bitcoin is freedom

SHA-256 hash

04a474a12bad5d61e9d461945e6785e28b8789c591efc33a75621a2e65183393

Ok great, so we can play silly games with Hash functions, but what else can we do?

Well hashes are used for all kinds of things in computer science.

One common use case is storing passwords!

Imagine if a website just stored your password, in plain text, in their database.

Any employee could easily see all the passwords. And if any attacker got access to the database, they'd get them all too!

Bad security!

Instead, websites store the *hashes* of passwords

When you login, the website takes your password and hashes it, then compares it to the hash that's stored in their database...

If it matches, you're good to go...

Ever wonder why the "Forgot my password" always makes you create a new password (instead of just telling you what your old one was)?

It's because the website doesn't know what your old password was!

They only have the Hash. They have no way of figuring out what the original was

Great so Hashes are neat and useful, but how do they relate to [#Bitcoin](#)? .....

In Bitcoin, the SHA-256 function plays a key role in addresses, address scripts, and mining.

I won't go into all the details here, but I will finish with one more game that will begin to explain the mechanics of how Bitcoin uses hashes in its mining system...

Ok here's the game...

I want you to come up with some data that starts with the word "Bitcoin", and SHA-256 hashes to something that begins with 000.

First person to come up with an answer wins a reward.

What's your strategy?

Well, remember that there's no way to start with the Hash and figure out the input data....

The only way is to brute force guess the input data!!

We'll brute force it like this:

Calculate the hash of "Bitcoin-1" and see if it starts with 000.

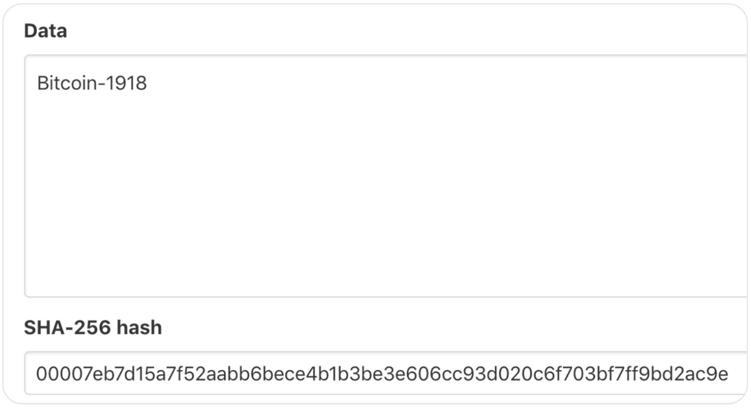
Calculate the hash of "Bitcoin-2" and see if it starts with 000.

Then try "Bitcoin-3", "Bitcoin-4", etc.

The only way is to keep trying and checking until you find one!

Eventually if you keep up this process, you'll come across the number 1918, and check it out!

"Bitcoin-1918" hashes to 00007eb7d15a7f52aabb6bece4b1b3be3e606cc93d020c6f703bf7ff9bd2ac9e, which starts with '000' !!



Data

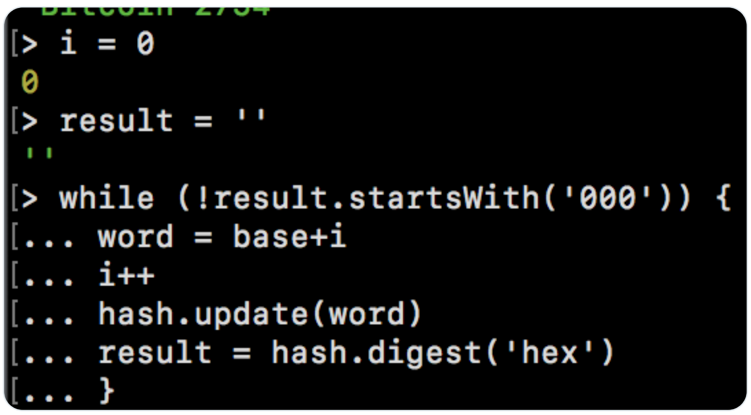
Bitcoin-1918

SHA-256 hash

00007eb7d15a7f52aabb6bece4b1b3be3e606cc93d020c6f703bf7ff9bd2ac9e

How did I find that out? Well, I just wrote a simple script that does that same brute force process.

Try one number, check it, move on to the next, try it, etc... until you find a winner...



```
[> i = 0
0
[> result = ''
''
[> while (!result.startsWith('000')) {
[... word = base+i
[... i++
[... hash.update(word)
[... result = hash.digest('hex')
[... }
```

In fact, what I've done is I "Proved" to you that my computer did some "Work"...

Does "Proof of Work" sound familiar?

Just edit our little game and replace the starting word "Bitcoin" with a list of bitcoin transactions thats basically how the bitcoin mining competition works!

Miners are all trying to find a list of valid of transactions that Hash to something with a bunch of leadings 0s

And when one of them finds a valid one, they get to add a block to the blockchain and claim a big reward

Anyway, that's the basics, and I'll end the thread on Hash Functions here.

Follow + DM or comment here if you have any questions about technical concepts that you'd like an overview of!

I’ve purposefully glazed over the specific details of mining - we’ll save those details for another thread :)

@aantonop - I think your followers might enjoy this beginner’s thread on Hash functions.

@jimmysong I think your followers might like this beginner’s thread on Hash functions

Thanks for everything you taught me in the Programming Bitcoin course!

• • •