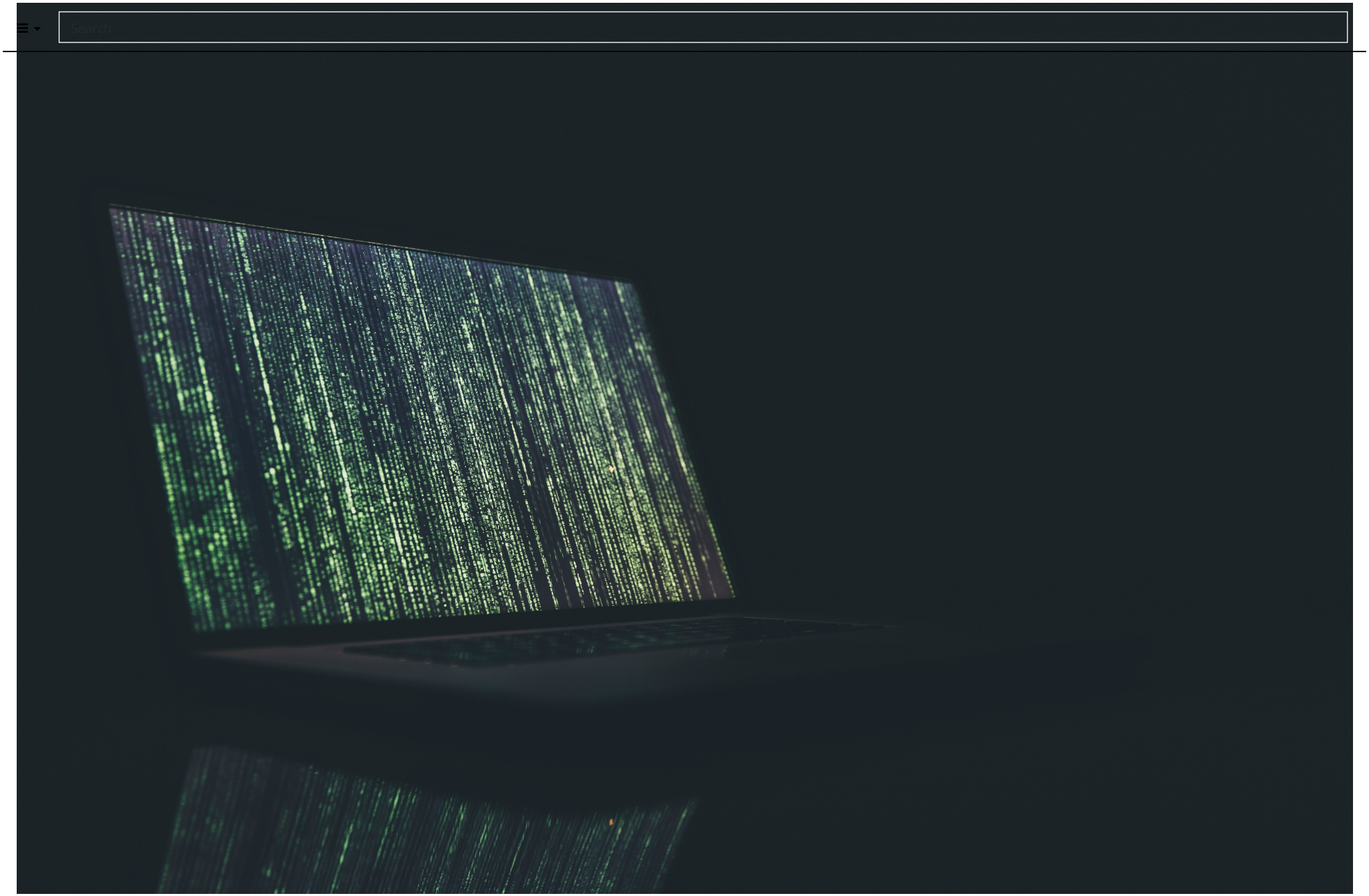
How Does a Hashing Algorithm Work?

16 comments | Antonio Madeira | 13 Mar 2019 | 60.75 k



Hashing algorithms are an important weapon in any cryptographer (/wallets/guides/how-does-bitcoin-cryptography-work/)'s toolbox. They are everywhere on the internet, mostly used to secure passwords, but they also make up an integral part of most cryptocurrencies such as Bitcoin (/coins/btc/overview) and Litecoin (/coins/ltc/overview).

The main feature of a hashing algorithm is that it is a one-way function – you can get the output from the input but you can't get the input from the output – just like elliptic curve cryptography (/wallets/guides/what-is-elliptic-curve-cryptography/) where you can't get the private key (/wallets/guides/what-is-a-bitcoin-private-key/) from the public key. The other key property is that the same input creates the same output.

Most hashing algorithms, including the SHA and RIPEMD are all descended from the MD4 family. The MD4 hashing algorithm was developed by Ronald Rivest specifically to allow very easy software implementation. The MD4 algorithm and subsequent SHA algorithms use 32 bit variables with bitwise Boolean functions such as the logical AND, OR and XOR operators to work through from the input to the output hash.

So how does a hashing algorithm work – in this case a look at SHA1:

1 – Create five variables

> H0 - 01100111010001010010001100000001
> H1 - 11101111110011011010101110001001
> H2 - 10011000101110101101110011111110
> H3 - 00010000001100100101010001110110
> H4 - 11000011110100101110000111110000

2- Then choose a word to hash. In this case we will choose the word “CRYPTO”

3- Convert the word to ASCII – “American Standard Code for Information Interchange”. Each letter has a number assigned to it.

CRYPTO – 67-82-89-80-84-79

4- Convert ASCII code to binary –

> CRYPTO – 01000011-01010010-01011001-01010000-01010100-01001111

5- join characters and add 1 to the end.

> CRYPTO – 0100001101010010010110010101000001010100010011111

6- Add zeros to make the message equal to 448 mod 512 – (modular arithmetic is just like a clock except with 512 hours). So a 48 bit message with the added one will need to have 399 zeros added to the end, and if the message was 64 characters (or 512 bits) long you would need 447 zeros.

> 010000110101001001011001010100000101010001001111100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000-
> 000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000-
> 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

7- Add the original message length into the 64 bit field left over after the 448 modular arithmetic. The message is 48 characters long which expressed in binary is 110000. So the below is added to the end of the message in part six.

0000000000000000000000000000000000000000000000000000000000110000

8- Break the message up into sixteen sections of 32 characters/bits.

01000011010100100101100101010000
01010100010011111000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000110000

9- Transform the 16 x 32 character bit words into 80 words using a step loop function. First select four words for the first run through the loop which are strings 1,3,9 &14 from step 8.

The next time through the loop we will use words 2,4,10,15 from stage 8.

The next process is to XoR the words together. Xoring is just a basic computational function that gives the output of q only if the two inputs both have a 1 in that position – if they don't the output is zero.

The function is ((14 XOR 9) XOR 3) XOR 1) which is:

00000000000000000000000000000000

XOR

01000011010100100101100101010000

Is

01000011010100100101100101010000

10- perform a left rotate on the numbers – i.e. move the left most digit to the right.

So

01000011010100100101100101010000

Becomes

> 10000110101001001011001010100000

This process is then repeated until there are 80 words, or strings of 32 bits.

10- The next step is to run a set of functions over the words in a specific order operating off the five variables that were set in step 1. The functions combine AND, OR & NOT operators combined with left shifts.

The end result is that you are left with five variables of:

> H0 – 01000100101010010111000100110011

> H1- 01010000111001010011100001011000

> H2-11110000010110000100011000111101

> H3-01001011111101111111000111100101

> H4-01000010110110011100101001001011

11- Convert the H variables into hex:

> H0- 44a97133

> H1- 50e53858

> H2- f058463d

> H3 - 4bf7f1e5

> H4 - 42d9ca4b

12- Join the variables together to give the hash digest:

> 44a9713350e53858f058463d4bf7f1e542d9ca4b

This is the basic process behind hashing – simply convert a number into binary then perform a set of simple functions that operate through basic standard transistor and bus processes such as AND, XOR, NOT, Rotate &OR. This is part of the reason that ASICs, or Application Specific Chips can be designed that optimize hashing. In the case of SHA-256 - chips have been specifically designed to optimize the iterations throughout the steps to increase the speed of creating a hash from an input. In the case of mining, this means you can calculate more hashes per second by iterating through the nonce and extra nonce parameters and have a higher probability of winning the block reward.

Comments

Search

What is your opinion on How Does a Hashing Algorithm Work? (You will be banned if you post affiliate / referral links. Please keep the conversation civilized and remember, we're all in this together!)

Post

sotcao9 ()
1 year ago
(/profile/sotcao9/)

Tham gia các biến với nhau để đưa ra thông báo băm:

44a9713350e53858f058463d4bf7f1e542d9ca4b

reply () Agree () ()

[removed] ()
2 years ago

Hashing algorithms are cryptographic hash functions that are based on hashing mechanism. They utilize the cryptographic hashing to secure the data through a hash. Its working process is quite complex, first it maps the data of random size and applies a hashing function to convert it into a hash code, where hash is an alphanumeric string that is cryptographic conversion of the data.

Basically, it is used for security purposes such as creating digital signature, file verification systems and other types of authentication. This technology is also utilized in blockchain technology for creating cryptocurrencies. MD5 is a popular hashing algorithm that is being utilized by a majority of developers for integrating security in different types of algorithm.

reply () Agree () ()

glisero_kalimero ()
2 years ago
(/profile/glisero_kalimero/)

Can anybody explain step 10 , it's looking like so complicated

reply () Agree () ()

esraa.essam1 ()
2 years ago
(/profile/esraa.essam1/)

in step 9, why strings 1,3,9 and 14 are chosen? what is the sequence for this selection?

reply () Agree () ()

sananaz735026 ()
2 years ago
(/profile/sana

this process is not feasible by hand... how can we make computer programm for it?

naz73526/) Search

reply () Agree () ()

virtue44 ()
3 years ago

(/profile/virtue44/)

It is very helpful to understand the algorithm, Thanks!

reply () Agree () ()

palmakgroup ()
3 years ago

(/profile/palmakgroup/)

I am a Turkish man. Before read your web page I have seen too much web pages in Turkish(my native language) and I can not understand them. when ı read your page I understood the algorithm. Many thanks

reply () Agree () ()

A18STE ()
3 years ago

(/profile/A18STE/)

For me this is kinda complex but the explanation is really good!

reply () Agree () ()

cgefter ()
3 years ago

(/profile/cgefter/)

The explanation is awesome- no need to know exact details but idea that variable input can result in fixed size output is clear and any change in input gives wildly different HASH and non-reversable is pretty darn clear-thanks

reply () Agree () ()

kovac.kornelije ()
3 years ago

(/profile/kovac.kornelije/)

And how do you get the original message back?

1 reply () Agree () ()

maazjawaid28 ()
3 years ago

(/profile/maazjawaid28/)

The explanation is quite helpful, But I can't find any clue that from where the strings 1,3,9 and 14 comes from (see step 9) ?

reply () Search Agree () ()

chalkiewa ()
3 years ago

(/profile/chalkiewa/)

Very interesting. How are the first five variables created - randomly? In step 9 why are strings 1, 3, 9, 14 chosen?

1 reply () Agree () 1 () ()

mouryavamsi.n ()
3 years ago

(/profile/mouryavamsi.n/)

This is INSANE math !!!

reply () Agree () ()

dfauchier ()
3 years ago

(/profile/dfauchier/)

Thank you SO much for explaining. Didn't really understand 8,9,10, but still got lots of value

reply () Agree () ()

TortyCash ()
3 years ago

(/profile/TortyCash/)

Thank you for this explanation!

reply () Agree () ()

acaciosc ()
4 years ago

(/profile/acaciosc/)

The part 11 should have been better explained. Like, for example... HOW the "H" variables became especifically those hex codes!

6 replies () Agree () 4 () ()

Search

Related guides

- What is a Bitcoin Fork? (/coins/guides/what-is-a-bitcoin-fork/)
- Bitcoin Transactions – Scriptsig and Scriptpubkey – locking and unlocking a transaction? (/coins/guides/bitcoin-transactions-scriptsig-and-scriptpubkey-locking-and-unlocking-a-transaction/)
- How does the Bitcoin Network actually work? (/coins/guides/how-does-the-bitcoin-network-actually-work/)
- How does a Bitcoin node verify a transaction? (/coins/guides/how-does-a-bitcoin-node-verify-a-transaction/)
- How Does Bitcoin Cryptography work? (/wallets/guides/how-does-bitcoin-cryptography-work/)
- What is a Merkle Tree? (/mining/guides/what-is-a-merkle-tree/)
- How to trade Bitcoin with the Relative Strength Index? (/exchanges/guides/how-to-trade-bitcoin-with-the-relative-strength-index/)

Latest guides

- Beyond the Token Economy – How Convergence Brings Private Investment Into DeFi (/coins/guides/beyond-the-token-economy-how-convergence-brings-private-investment-into-defi/)

Important information

This website is only provided for your general information and is not intended to be relied upon by you in making any investment decisions. You should always combine multiple sources of information and analysis before making an investment and seek independent expert financial advice.

Where we list or describe different products and services, we try to give you the information you need to help you compare them and choose the right product or service for you. We may also have tips and more information to help you compare providers.

Some providers pay us for advertisements or promotions on our website or in emails we may send you. Any commercial agreement we have in place with a provider does not affect how we describe them or their products and services. Sponsored companies are clearly labelled.

Search

Privacy policy (/privacy-policy/)

Terms & Conditions (/terms-conditions/)

Website disclaimer (/website-disclaimer/)

Submit Content (https://drive.google.com/open?id=0B3isfML9O09eMHQ2dGFPOHBSYlU)

Developers

Widgets (/dev/widget/wizard/)

</> API (https://min-api.cryptocompare.com)

Turn Lights Off ()

Conference Screen (https://www.cryptocompare.com/full-screen/conference?fsyms=BTC,ETH,BCH&tsym=USD&eventLogo=https://www.cryptocompare.com/media/30001628/your-logo.png)

Conference Ticker (https://www.cryptocompare.com/media/25792605/topcoinsticker-v3.html)

Get in touch

Careers (/careers/)

Contact Us (https://cryptocompare.zendesk.com/hc/en-gb/requests/new)

FAQs (https://cryptocompare.zendesk.com/hc/en-gb)

Get the CryptoCompare App

ANDROID APP ON Google play (https://play.google.com/store/apps/details?id=com.cryptocompare.mainapp)

Download on the App Store (https://itunes.apple.com/us/app/cryptocompare/id1248404900?ls=1&mt=8)

Made with ♥ in London (/about-us/)