

Immutable data

Description

default watermark

An application on the exorbin.com website can convert any block of data into a near unique 64 character string of characters.

Fun with SHA256

So â??abcâ?• converts to
ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad

My name, â??Richard Coyneâ?•, converts to
ba7c5c9e90f05540084e798effd0389b60ec98677662017065d34a0b880778a7

Hereâ??s some Shakespeare

> â??To be, or not to be â?? that is the question:
> Whether â??tis nobler in the mind to suffer
> The slings and arrows of outrageous fortune
> Or to take arms against a sea of troubles,
> And by opposing end them. To die â?? to sleep â??
> No more; and by a sleep to say we end
> The heartache, and the thousand natural shocks
> That flesh is heir to. â??Tis a consummation
> Devoutly to be wishâ??d. To die â?? to sleep.
> To sleep â?? perchance to dream: ay, thereâ??s the rub!â?•

That translates to
ff5397a0736b5a58b3e4a5950982296967ce1c9ad26c4277b0094acbb3196edb

The arbitrary looking 64 character string output serves as a fixed-size unique condensed representation of the original data. Take out the first comma in the Shakespeare above and you end up with a very different string output

e1ad69b86b1e897c34ad111959543515f2f5b729bf359646b3fad999d2a3b24c

The conversion is not a method for compressing or encrypting data. Thereâ??s no algorithm to unpack the 64 character string to recover the original data. The output string is for machine consumption rather than for human beings to read, check and ponder.

The output string is called a *hash*, and the SHA256 algorithm used by that website is the latest, the most sophisticated and the least likely to generate â??a collision,â?• i.e. two or more sets of different data that have the same hash. The conversion provided by the exorbin.com website application is just a demonstrator of processes normally invisible to users.

Why hash?

What is the use of this (near) unique hash representation of a block of data? Itâ??s used for authentication. As well as transmitting the segment of Shakespeare, transmit the hash as well â?? perhaps in a different channel. The receiver software then runs the data through the SHA256 algorithm, and if the receiver generated hash does not match the hash sent with the data, then that alerts the receiver that the data has been tampered with en route. It wonâ??t tell you what was changed, or the magnitude, but it may send up an alert, or notify the sender software to transmit the data again.

default watermark

A hash can be used to transmit a password over the Internet. My online banking service has my password in a database saved as a hash. So if I enter my (not really) password â??appLeTr33â?• via the banking website then the transmission software converts that to the hash
80dfcee4ead6ef8cd1ba4edfc782d5c4b4505affc223d1345bcce51c9818fc61
and sends the hash through the Internet. The server software receives the hash and looks that up to access my account. If the hash fell into the wrong hands then there is almost no way to reverse engineer the hash string to the original password. So thatâ??s one of several measure for keeping passwords safe in transit.

Hacking ledgers

Another use of the SHA256 hash algorithm is to retain the integrity of a ledger of data, e.g. a sequence of monetary transactions.

Hereâ??s a page from a spreadsheet of transactions. Iâ??ve added a column to show the SHA256 hash of the data in each row, added to the hash of the previous row. Yes, you can have a hash of a hash.

seller	buyer	item	amount	combined	hash
sue	fred	vase	54.22	suefredvase54.220	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7
paul	val	chair	150.00	paulvalchair150.00e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	77bc0219dff818d4f1f1a17077f75885d2e7c5fcda47d39dd3ead8eef6
sue	mike	desk	32.00	suemikedesk32.0077bc0219dff818d4f1f1a17077f75885d2e7c5fcda47d39dd3ead8eef6dec7f0	674833caff6c2777543aa705f2fe34d59a51489c83d7bafacfcce04516

If someone changes the content of one of the rows in the ledger then that changes the hash values for the transactions that follow, as shown here.

seller	buyer	item	amount	combined	hash
sue	fred	vase	54.22	suefredvase54.220	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7
paul	val	chair	200.00	paulvalchair200.00e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	16f9fd11459ac305e6eade9b4d56e87e9b3500a6aa14f2ca335264c8
sue	mike	desk	32.00	suemikedesk32.0016f9fd11459ac305e6eade9b4d56e87e9b3500a6aa14f2ca335264c88cf25de5	4cc5f86799bd1996b84c69cacc034f91058ce6c7ed2a0a92a724e6f06

With access to the SHA256 algorithm, someone who wanted to doctor the ledger could change the hash value of that transaction and those that follow. But if the ledger has already been shared with others and verified, then the discrepancy will be obvious, and rejected as invalid by an auditing algorithm.



Rock solid

In fact, this method of verification via hashing works on whole pages of transactions as well as for individual transactions. One of the best descriptions of this method of verification that I have found comes from a blog by Antony Lewis.

A traditional paper ledger consists of pages of transactions. Instead of page numbers at the top of each page (header) print a hash of the previous page. At the bottom of the page print a hash of the page (including its header). Lewis calls the hash a unique fingerprint of the page, the page is a *block*, the whole set of pages linked in this way is a *chain*.

> â??By using a fingerprint instead of a timestamp or a numerical sequence, you also get a nice way of validating the data. In any blockchain, you can generate the block fingerprints yourself by using some algorithms. If the fingerprints are consistent with the data, and the fingerprints join up in a chain, then you can be sure that the blockchain is internally consistent. If anyone wants to meddle with any of the data, they have to regenerate all the fingerprints from that point forwards and the blockchain will look different.â?•

The term *immutable* is used commonly to describe this state of a data set. Each page of transactions gets embedded in the transactions that follow. So you end up with a massive unalterable data set. Itâ??s a bit like sedimentary rock formations where the newest strata depend for their integrity on older formations below, which in turn require more effort to penetrate (i.e. alter).

This â??immutabilityâ?• is a key feature of distributed digital ledgers as used in cryptocurrencies, where the ledger is duplicated and shared across a large number of users, and there is no centralised keeper of the data, or auditing authority such as a bank. Also see my posts Digital money and Wasting time in the bit economy.

References

- Lewis, A. (2015), â??A Gentle Introduction to Blockchain Technologyâ??. *Bits on blocks: Thoughts on blockchain technology*, 9 September. Available online: https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/ (accessed 28 July 2017).
- Lewis, A. (2016), â??A Gentle Introduction to Immutability of Blockchainsâ??. *Bits on blocks: Thoughts on blockchain technology*, 29 February. Available online: https://bitsonblocks.net/2016/02/29/a-gentle-introduction-to-immutability-of-blockchains/ (accessed 28 July 2017).

Note

- Picture is of South Stack, Holy Island, Wales.

Category

1. Economics

default watermark

Tags

1. Big Data
2. bitcoin
3. blockchain
4. ciphercity
5. cryptocurrency

Date Created
July 29, 2017
Author
rcoyne99