Study of Crypto Currency and Comparison of Two Algorithms for the Crypto Currency

[1]Rohini S Gaikwad, [2]Tanmay Pimple, [3]Divya Premchandran

[1]Student, [2]Student, [3]Assistant Professor
[1]Master of Computer Applications,
[1]Bharati Vidyapeeth's Institute of Management and Information Technology, Belapur (CBD), India

Abstract : In today’s situation, everything is getting advanced. It is the period of the digitalization. Every single work is getting digitalised, from Exchanging to E-Learning. In the period of the digitalisation, the exchanges are made in the advanced way. Cryptographic money is the computerized cash which is created in the carefully and utilized as advanced instalment through the advanced wallet. Digital currency is created by Mr David Chaum. One of the digital currency is named as Bitcoin. Bitcoin is the principal computerized money which came in the market in the year 2008. Digital currency has made the distributed exchanges and this money is all around acknowledged. Increasingly the money is utilized and the estimation of the cash increments. The point for the exploration paper is to discover the contrast between two calculations. SHA256 and Scrypt.

***IndexTerms* - Scrypt, SHA-256, CryptoCurrecncy, Bitcoin**

I. Introduction

In this we are comparing two algorithm SHA-256 and scrypt for cryptocurrency, which is used by Bitcoin (SHA-256) and litecoin(scrypt). A cryptocurrency is a digital asset designed to work as a medium of exchange that uses strong cryptography to secure financial transactions, control the creation of additional units, and verity the transfer of assets. Cryptocurrency is a kind of digital currency, vitual currency, or alternative currency.

The principal decentralized digital money, bitcoin, was made in 2009 by pseudonymous engineer Satoshi Nakamoto. It utilized SHA-256, a cryptographic hash work, as its evidence of-work plot. In April 2011, Namecoin was made as an endeavor at framing a decentralized DNS, which would make web oversight extremely troublesome. Before long, in October 2011, Litecoin was discharged. It was the principal fruitful digital currency to utilize scrypt as its hash work rather than SHA-256. Another remarkable digital money, Peercoin was the first to utilize a proof-of-work/verification of-stake half breed.

II. Literature Review

SHA256

SHA256 is a cryptographic hash work. In that capacity it is for all intents and purposes difficult to invert it and discover a message that hashes to a given process.

Also, it is extremely effective. This is for the most part something to be thankful for.

Be that as it may, a standout amongst the most understood employments of a hash work is watchword hashing: putting away the hash of a secret word rather than the real watchword itself. In the wake of writing in the secret word, it gets hashed and contrasted with the put away process. This is more secure than putting away the watchword free.

Yet, imagine a scenario where a hashed secret word spills out. We can play out a word reference assault: simply attempt each secret key we can consider, hash it, at that point contrast it with the spilled process. Most passwords are from a sensibly little rundown. Abruptly the productivity of SHA256 isn't so awesome any longer, since it enables us to attempt numerous passwords in a brief span.

This is the place secret word based key deduction capacities like Scrypt and companions come in. They are capacities that have an indistinguishable property from a hash work (and in light of them) yet rather are extremely moderate and memory wasteful. While this doesn't make a difference for a solitary check, it truly begins to tally when you need to attempt billions of passwords.

With SHA-256, you require just processing energy to run the calculation. Sufficiently given registering power, you can ascertain SHA-256 rapidly.

Scrypt

Scrypt, then again, utilizes registering power as well as memory. This is on the grounds that it produces a considerable measure of pseudo-irregular information, stores that in memory, and after that references that information in a random(*) way. Scrypt at that point utilizes this information in producing the hash. This implies scrypt requires both expanding memory and figuring power.

III. The litecoin digital currency

- The Litecoin (LTC) computerized money, one of the primary Alt-Coin frameworks, was propelled in October 2011, and is viewed as the best of the considerable number of monetary forms created through the "forking" process, got from the Bitcoin framework.
- As in the Tenebrix framework's case, from which it determined, Litecoin utilizes a proof-of-work (POW) then again calculation, called "scrypt", a capacity that requires a measure of RAM (Random Access Memory) so as to be

processed. Another development brought by the Litecoin framework lies in the way that it has actualized a speedier square age strategy that lessens the computational time and accelerates the exchanges' incorporation in a square.

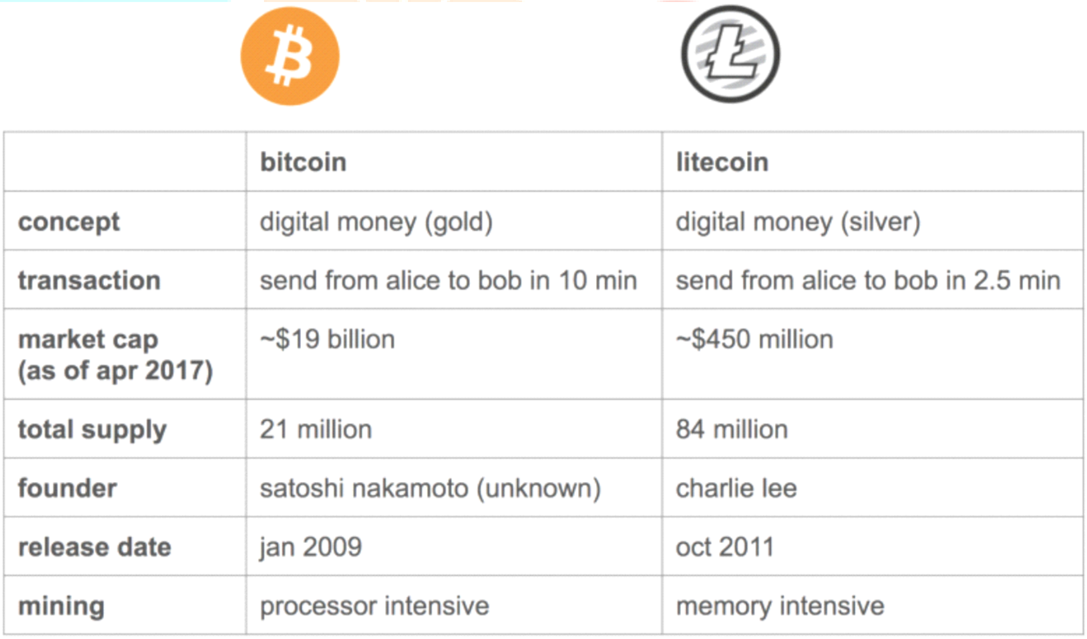
- The "scrypt" produces pseudorandom numbers that are put away in the Random Access Memory, with a specific end goal to be additionally gotten to. The Litecoin's calculation gets to this memory a few times with a specific end goal to restore the outcome. The Litecoin's "scrypt" execution requires just 128 kB of memory, a sum that ought not raise asset issues for the PCs of the framework's hubs. Utilizing a memory-hard capacity as a proof-of-work has certain points of interest. To begin with, the quantity of excavators increments as any individual who possesses a PC can play out this errand. On account of Bitcoin, mining requires a particular hardware and this reality lessens the quantity of diggers. Another favorable position of the "scrypt" framework is that in a portion of the cases it can bring down the asset squander contrasted with the customary verification of- work frameworks .
- The Litecoin devotees think about that a standout amongst the most imperative preferences of this framework is the way that it actualizes the "scrypt" that can maintain more assaults made through Graphics Processing Units (GPUs) and Application-Specific Integrated Circuits (ASICs) than the SHA-256 calculation actualized in the Bitcoin framework .

IV. SHA-256 Versus Scrypt

- SHA-256 and Scrypt are the two extremely normal calculation frameworks which are utilized by cryptographic money excavators to check squares of exchange information. The framework utilized, is chosen by the engineers of the separate cryptographic money. These two calculations are constantly incorporated into every digital money talk.

- Bitcoin uses the SHA-256 algorithm to “mine” new coins, leading to giant setups that use tons of specially designed mining hardware to crank out more coins. Litecoin, on the other hand, uses Scrypt, which demands memory instead of processor resources. That stops these giant setups from easily switching to Litecoin.

- The currency is also meant to be faster than Bitcoin, leading to the comparison of /Bitcoin as gold just as Litecoin is to silver. Where it takes ten minutes to log a Bitcoin transaction into the blockchain, it takes just two and a half minutes for Litecoin to do the same.



	bitcoin	litecoin
concept	digital money (gold)	digital money (silver)
transaction	send from alice to bob in 10 min	send from alice to bob in 2.5 min
market cap (as of apr 2017)	~$19 billion	~$450 million
total supply	21 million	84 million
founder	satoshi nakamoto (unknown)	charlie lee
release date	jan 2009	oct 2011
mining	processor intensive	memory intensive

Figure 1 : Comparison of bitcoin and litecoin

- Bitcoin’s transaction confirmation time is 10 minutes while Litecoin’s is only 2.5 minutes. Litecoin is able to process a higher volume of transactions due to the faster transaction confirmation time.

- Bitcoin mining uses the algorithm SHA-256 which is processor intensive while Litecoin mining uses Scrypt which is more memory intensive. Early on Bitcoin was able to be mined using redular computers (CPUs) and later on more powerful gaming computers. (GPUs). Now due to the increase in mining difficulty created by competition within the network the only way mining is profitable after accounting for electricity and mining equipment costs is through the use of Application Specific Integrated Circuit (ASIC) miners. These are expensive machines built specifically for mining and won’t be able to be used for anything else. Since only some people have the resources to buy and operate ASICs, Charlie did not want Litecoin mining to be dominated by ASICs. Therefore he created Litecoin with a more memory intensive mining algorithm to make mining Litecoin less efficient for ASICs and was more accessible to everyone.

V. SHA-256 And Scrypt Algorithm

1. SHA-256 Algorithm

The cryptographic hash function SHA-256

General description

SHA-256 (secure hash algorithm, FIPS 182-2) is a cryptographic hash function with digest length of 256 bits. It is a keyless hash function; that is, an MDC (Manipulation Detection Code).
A message is processed by blocks of 512 = 16 × 32 bits, each block requiring 64 rounds.

Basic operations

- Boolean operations AND, XOR and OR, denoted by $\wedge$, $\oplus$ and $\vee$, respectively.
- Bitwise complement, denoted by ¯.
- Integer addition modulo 232, denoted by A + B.

Each of them operates on 32-bit words. For the last operation, binary words are interpreted as integers written in base 2.

- RotR(A, n) denotes the circular right shift of n bits of the binary word A.
- ShR(A, n) denotes the right shift of n bits of the binary word A.
- AkB denotes the concatenation of the binary words A and B.

Functions and constants

The algorithm uses the functions:
$Ch(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z)$,
$M\ aj(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z)$,
$\Sigma0(X) = RotR(X, 2) \oplus RotR(X, 13) \oplus RotR(X, 22)$,
$\Sigma1(X) = RotR(X, 6) \oplus RotR(X, 11) \oplus RotR(X, 25)$,
$\sigma0(X) = RotR(X, 7) \oplus RotR(X, 18) \oplus ShR(X, 3)$,
$\sigma1(X) = RotR(X, 17) \oplus RotR(X, 19) \oplus ShR(X, 10)$,
and the 64 binary words Ki given by the 32 first bits of the fractional parts of the cube roots of the first 64 prime numbers:
0x428a2f98 0x71374491 0xb5c0fbcf 0xe9b5dba5 0x3956c25b 0x59f111f1 0x923f82a4 0xab1c5ed5
0xd807aa98 0x12835b01 0x243185be 0x550c7dc3 0x72be5d74 0x80deb1fe 0x9bdc06a7 0xc19bf174
0xe49b69c1 0xefbe4786 0x0fc19dc6 0x240ca1cc 0x2de92c6f 0x4a7484aa 0x5cb0a9dc 0x76f988da
0x983e5152 0xa831c66d 0xb00327c8 0xbf597fc7 0xc6e00bf3 0xd5a79147 0x06ca6351 0x14292967
0x27b70a85 0x2e1b2138 0x4d2c6dfc 0x53380d13 0x650a7354 0x766a0abb 0x81c2c92e 0x92722c85
0xa2bfe8a1 0xa81a664b 0xc24b8b70 0xc76c51a3 0xd192e819 0xd6990624 0xf40e3585 0x106aa070
0x19a4c116 0x1e376c08 0x2748774c 0x34b0bcb5 0x391c0cb3 0x4ed8aa4a 0x5b9cca4f 0x682e6ff3
0x748f82ee 0x78a5636f 0x84c87814 0x8cc70208 0x90befffa 0xa4506ceb 0xbef9a3f7 0xc67178f2

Padding

To ensure that the message1 has length multiple of 512 bits:

- first, a bit 1 is appended,
- next, k bits 0 are appended, with k being the smallest positive integer such that $l + 1 + k \equiv 448$ mod 512, where l is the length in bits of the initial message,
- finally, the length $l < 2$

64 of the initial message is represented with exactly 64 bits, and these bits are added at the end of the message.
The message shall always be padded, even if the initial length is already a multiple of 512.

Block decomposition

For each block $M \in \{0, 1\}$
512, 64 words of 32 bits each are constructed as follows:

- the first 16 are obtained by splitting M in 32-bit blocks

M = W1kW2k · · · kW15kW16

- the remaining 48 are obtained with the formula:

$Wi = \sigma1(Wi-2) + Wi-7 + \sigma0(Wi-15) + Wi-16, 17 \leq i \leq 64$.
Hash computation

- First, eight variables are set to their initial values, given by the first 32 bits of the fractional part of the square roots of the first 8 prime numbers:

H(0)
1 = 0x6a09e667 H(0)
2 = 0xbb67ae85 H(0)
3 = 0x3c6ef372 H(0)
4 = 0xa54ff53aH(0)
5 = 0x510e527f H(0)
6 = 0x9b05688c H(0)
7 = 0x1f83d9ab H(0)
8 = 0x5be0cd19

• Next, the blocks M(1), M(2), . . . , M(N) are processed one at a time:
For t = 1 to N
– construct the 64 blocks Wi from M(t), as explained above
– set
(a, b, c, d, e, f, g, h) = (H(t−1)1, H(t−1)2, H(t−1)3, H(t−1)4, H(t−1)5 , H(t−1)6, H(t−1)7, H(t−1)8)
– do 64 rounds consisting of:
T1 = h + Σ1(e) + Ch(e, f, g) + Ki + Wi
T2 = Σ0(a) + M aj(a, b, c)
h = g
g = f
f = e
e = d + T1
d = c
c = b
b = a
a = T1 + T2
1 We assume that the length of the message can be represented by a 64-bit integer.
2 – compute the new value of H(t)j H(t)
1 = H(t−1)1 + aH(t)
2 = H(t−1)2 + b H(t)
3 = H(t−1)3 + c H(t)
4 = H(t−1)4 + d H(t)
5 = H(t−1)5 + e H(t)
6 = H(t−1)6 + f H(t)
7 = H(t−1)7 + g H(t)
8 = H(t−1)8 + h
End for
• The hash of the message is the concatenation of the variables HNi
after the last block has been processed H = H(N)1
kH(N)2
kH(N)3
kH(N)4
kH(N)5
kH(N)6
kH(N)7
kH(N)8.
Implementation: signatures

Implement the cryptographic hash function just described. Define the class sha256 with the method: public static BigInteger hash(byte[] M)
input: M is a chain of bytes of arbitrary length;
output: a positive integer in the interval [0, 2256), the value of the hash of M.
Test values
To check the implementation, you can use the following values, given in hexadecimal notation.
input 61 62 63
hash ba7816bf 8f01cfea 414140de 5dae2223 b00361a3 96177a9c b410ff61 f20015ad
input 61 62 63 64 62 63 64 65 63 64 65 66 64 65 66 67 65 66 67 68 66 67 68 69 67
68 69 6a 68 69 6a 6b 69 6a 6b 6c 6a 6b 6c 6d 6b 6c 6d 6e 6c 6d 6e 6f 6d 6e 6f 70 6e 6f 70 71
hash 248d6a61 d20638b8 e5c02693 0c3e6039 a33ce459 64ff2167 f6ecedd4 19db06c1
input One million of 61
hash cdc76e5c 9914fb92 81a1c7e2 84d73e67 f1809a48 a497200e 046d39cc c7112cd0

2. Scrypt Algorithm :

The algorithm includes the following parameters:

Passphrase - The string of characters to be hashed.
Salt - A string of characters that modifies the hash to protect against Rainbow table attacks
N - CPU/memory cost parameter.
p - Parallelization parameter; a positive integer satisfying p = (232- 1) * hLen / MFLen.
dkLen - Intended output length in octets of the derived key; a positive integer satisfying dkLen = (232- 1) * hLen.
r - The blocksize parameter, which fine-tunes sequential memory read size and performance. 8 is commonly used.
hLen - The length in octets of the hash function (32 for SHA256).
MFlen - The length in octets of the output of the mixing function (SMix below). Defined as r * 128 in RFC7914.
Function Scrypt
Inputs:

Passphrase	:	Bytes string of characters to be hashed
Salt	:	Bytes random salt
CostFactor (N)	:	Integer CPU/memory cost parameter

BlockSizeFactor (r) : Integer blocksize parameter (8 is commonly used)
ParallelizationFactor (p) : Integer Parallelization parameter. (1..232-1 * hLen/MFlen)
DesiredKeyLen : Integer Desired key length in bytes
Output:
DerivedKey : Bytes array of bytes, DesiredKeyLen long

Step 1. Generate expensive salt blockSize ? 128*BlockSizeFactor //Length (in bytes) of the SMix mixing function output (e.g. 128*8 = 1024 bytes)
Use PBKDF2 to generate initial 128*BlockSizeFactor*p bytes of data (e.g. 128*8*3 = 3072 bytes)
Treat the result as an array of p elements, each entry being blocksize bytes (e.g. 3 elements, each 1024 bytes)
[B0...Bp-1] ? PBKDF2HMAC-SHA256(Passphrase,Salt, 1, blockSize*ParallelizationFactor)
Mix each block in B 2CostFactor times using ROMix function (each block can be mixed in parallel)
for i ? 0 to p-1 do
Bi ? ROMix(Bi, 2CostFactor)
All the elements of B is our new "expensive" salt expensiveSalt ? B0?B1?B2? ... ?Bp-1 //where ? is concatenation
Step 2. Use PBKDF2 to generate the desired number of bytes, but using the expensive salt we just generated return PBKDF2HMAC-SHA256(Passphrase, expensiveSalt, 1, DesiredKeyLen);

VI. CONCLUSION

Coinciding for a long time, Bitcoin and Alternative Coins were contending in a few angles, however the vast majority of the Lite-Coins monetary standards don't speak to an immediate contender of the Bitcoin. From the opposite, Bitcoin and Lite-Coins coordinate in a commonly helpful relationship. Bitcoin has profited from having the capacity to make the principal move with respect to the system impact and the liquidity. Likewise, Bitcoin has figured out how to cover altogether the clients' interest for computerized showcase monetary forms.

Nobody can state with conviction what will occur later on with these favourable circumstances, if Bitcoin will have the capacity to keep up its authority position or on the off chance that it will be surpassed as far as ubiquity, of the specialized or financial highlights by an effectively existing Litecoin or by one that will be produced later on. One thing is without a doubt: the overall acknowledgment of the computerized monetary forms could change the worldwide economy as we probably am aware it. Breaking down the development and attributes of the Bitcoin and Lite Coins, one can presume that their overall acknowledgment is not any more a unimaginable result, being a conceivable situation in what's to come.

Similarly as it occurred in the previous decades with the PCs and Internet, the effect of these computerized monetary standards will slowly increment later on, prompting real changes in our way of life, rethinking our regular day to day existence, economy and society.

VII. REFERENCES

[1]https://blog.coinbase.com/a-beginners-guide-to-litecoin-d9b455d44cd3

[2]https://en.wikipedia.org/wiki/Scrypt

[3]https://www.google.co.in/url?sa=t&source=web&rct=j&url=https://bitcoin.org/bitcoin.pdf&ved=2ahUKEwjdnp7TpPDbAhVLQY8KHVkJAzYQFjAAegQIBBAB&usg=AOvVaw05-4mYD7EyyKjwcHh8i0Vw

[4]https://medium.com/@rilcoin/cryptocurrency-hash-and-the-difference-between-sha-and-scrypt-1f2217eb5b89