# Bitcoin Difficulty, A Security Feature

Abdenaby Lamiri(✉), Kamal Gueraoui, and Gamal Zeggwagh

Modeling and Simulation in Mechanics and Energetics Team (MSME)
of the Research Center on Energy, Mohamed 5th University, Rabat, Morocco
abdlamsic@gmail.com, kgueraoui@yahoo.fr,
gamalzeggwagh972@hotmail.com

**Abstract.** Bitcoin has been growing, since its inception in 2009, to gain a financial mainstream despite the constant fluctuations in its value. It is currently ranked as the most successful Crypto-Currency and decentralized payment system among the others. This success is due, to some extent, to its security, which depends mainly on the cutting-edge cryptographic innovations, such as the hashing functions, the elliptic curve digital signature (ECDSA), and the difficulty that regulates the mining process and allows the system to keep up with the increasing hash-rate. This paper provides an overview of Bitcoin difficulty and how it contributes to the security of this Crypto-Currency.

**Keywords:** Bitcoin · Blockchain · Crypto-Currency · Difficulty
Security

## 1 Introduction

Since 2009, Bitcoin value has been soaring in a rapid rate until it reached a peak on December 17th, 2017 [1] when it attained, for the first time in history, more than $20,000 US dollars. Since then, its value has decreased tremendously, and it is currently fluctuating around $7, 000 US dollars for one bitcoin.

Bitcoin is a decentralized system that does not rely on any third party to process the transactions. Transactions are collected and validated by all the participating nodes connected to the peer-to-peer network. The validated transactions are stored in blocks and these blocks are added to the Blockchain. The Blockchain is a distributed ledger that contains all the valid transactions that ever happened in the system.

Bitcoin is considered a secure system since it relies on the implementation of some of the advanced cryptographic features. For instance, the Bitcoin keys and addresses generation process is secure because of the randomness used in producing the private keys, the elliptic curve discrete logarithm, which cannot be solved giving the current computational power, the one-way property of the hashing functions SHA256 and RACE MD (RIPEMD160), and because no backdoors were yet discovered in the elliptic curve. This elliptic curve used in Bitcoin is defined by a standard known as SECP256K1 [2].

This security is improved using the Base58Ckeck formatting, which ensure the integrity of the Data, mainly for the Bitcoin keys and addresses [3]. Also, the use of the ECDSA ensures that only the holders of the private keys can redeem the related funds.

Á. Rocha and M. Serrhini (Eds.): EMENA-ISTL 2018, SIST 111, pp. 367–372, 2019.
https://doi.org/10.1007/978-3-030-03577-8_41

Other security features are added to Bitcoin to ensure the most prominent security objectives such as the confidentiality, the integrity and the availability. These features are, for instance, the back-linkage of blocks, which helps ensure the integrity of the Blockchain; the Merkle tree, which ensures the Block integrity; and the difficulty, which ensures the system integrity since it forces miners to work hard for at least 10 min to find a proof-of work for a new block.

This paper aims to provide some insights about the Bitcoin difficulty by illustrating how it is related to the target and the Bits value, and how it contributes to the Bitcoin security. It provides some scripts, written in python 3, to calculate the difficulty and the target and to verify the proof-of-work. It shows also the correlation between the difficulty and the hashing rate used in Bitcoin.

## 2 Related Works

Since its inception in 2009, Bitcoin security became an active research area that interested many researchers around the world. Juan Garay, Aggelos Kiayias, and Nikos Leonardos studied Bitcoin difficulty and suggested a Bitcoin protocol with chains of variable difficulty as a way to deter any malicious adversary controlling a fraction of miners holding around 50% of the mining power [4]. Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse mentioned that the difficulty provides some resilience to mining power variation by allowing different instances of blockchain to tune their proof of work difficulty at different rate in order to maintain a stable rate of Blocks [5].

## 3 Bitcoin Difficulty

Difficulty can be defined as a measure of how difficult it is to find a hash (proof-of-work) below a given target [6]. This parameter is set dynamically by the Bitcoin network every 2016 blocks or two-weeks in average. The difficulty is tied to two other parameters, the target and the Bits, which we will explain in the following paragraphs.

The proof-of-work serves as a proof that the miner has committed a great amount of hashing power to find the block header's hash that satisfies the required condition. The proof-of-work is hard to find but easy to verify. It involves finding a value for the nonce that results in a block's header hash, using SHA-256 algorithm, that is less or equal to the difficulty target (target). So how this target is calculated?

Every block contains a field called "Bits", known also as target Bits, which is a four-byte number represented in a hexadecimal floating-point format. Bits value serves to calculate the difficulty target, which is used as a condition in the mining algorithm. The Bits field value of the first block in the Blockchain is 1d00ffff [7]. By convention, the first two digits (1d) represent the total number of digits a target is made of. It is used in the exponent of the floating-point notation while the remaining digits (00ffff) represents the coefficient. Now, how the target is derived from the Bits value?

To calculate the target from the Bits value, we rely on the following formula:

$$\text{TARGET} = \text{COEFFICIENT} * 2 ** (8 * (\text{EXPONENT} - 3)) \tag{1}$$

Where:

- *COEFFICIENT* is the three Bytes on the right part of 4-Byte format of the Bits.
- *EXPONENT* is the first Byte on the left part of 4-Byte format of the Bits.

Using the hexadecimal representation and applying this formula to the block #0 with Bits value of (0x1d00ffff), the target would be:

$$\text{TARGET} = \text{0X00FFFF} * 2 ** (\text{0X8} * (\text{0X1D} - \text{0X3})) \tag{2}$$

Therefore, the result in hexadecimal format is:

$$\begin{aligned}&\text{TARGET (in HEX)} = \\ &\text{0xffff000000000000000000000000000000000000000000000000}\end{aligned}$$

We compare the header's hash of the Block #0 (proof-of-work of Block #0) with the calculated target, using python 3. The following Script shows that the Block Header's Hash is less or equal the calculated target, which means that the proof-of-work (POW) is valid.

```
  >>> #calculating the target of Block #0 using the
Bits Value
>>> Target = 0x00ffff*2** (0x8*(0x1d-0x3))
>>> # the decimal number is:
>>> print (Target)
2695953529101130949315647634472399133601089873857416408
6137773096960
>>> # the Target in hexadecimal representation
>>> hex(Target)
'0xffff000000000000000000000000000000000000000000000
0000'
>>> # Now let's compare this target to the block #0
header's hash
>>>BlockHash=0x000000000019d6689c085ae165831e934ff763ae
46a2a6c172b3f1b60a8ce26f
>>> BlockHash <= Target
True
```

The target condition sets the frequency at which a new proof-of-work is found. It determines also the difficulty for a collection of blocks. Since the computational power is increasing at a rapid speed and the Bitcoin network must keep the block generation time at 10 min in average, the target should adjust accordingly. The retargeting is happening dynamically on every full node independently for every 2016 blocks, which occurs every two weeks. The retargeting formula used by Bitcoin full nodes is [8]:

$$\text{NEW TARGET} = \text{CURRENT TARGET} * (\text{TIME ON MINUTES OF THE LAST 2016 BLOCKS})/20160\ \text{MINUTES}. \tag{3}$$

The difficulty is tightly linked to the target and shows how it is difficult to find a new hash of a block that satisfies the target condition. Its main purpose is to regulate the mining process, so a new block is mined every 10 min in average. It is calculated using the following formula [6]:

$$\text{DIFFICULTY} = \text{TARGETMAX}/\text{TARGETCURRENT} \tag{4}$$

Where:

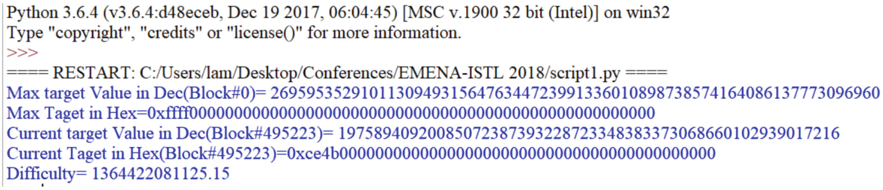
- *TargetMax* is the target of the genesis block (Block#0)
- *TargetCurrent* is the target of the current block

The following script is used to calculate the difficulty of a Block using its target and the target of the genesis block. We used the block #495223, mined on Nov 20, 2017 10:53:40 AM, to verify this script.

```
#calculating the target of Block #0 as the Target Max
using its Bits value
Target_Max = 0x00ffff*2** (0x8*(0x1d-0x3))
# the Max target in decimal number is:
print ("Max target Value in Dec(Block#0)=", Target_Max)
# the Max Target in hexadecimal representation
print("Max Taget in Hex="+hex(Target_Max))
#calculating the target of Block #495223 using the Bits
Value
Target_Current = 0x00ce4b*2** (0x8*(0x18-0x3))
# the current target in decimal number is:
print ("Current target Value in Dec(Block#495223)=",
Target_Current)
# the Current Target in hexadecimal representation
print("Current Taget in
Hex(Block#495223)="+hex(Target_Current))
# Calculating the Difficulty
print("Difficulty=",
round(Target_Max/Target_Current,2))
```

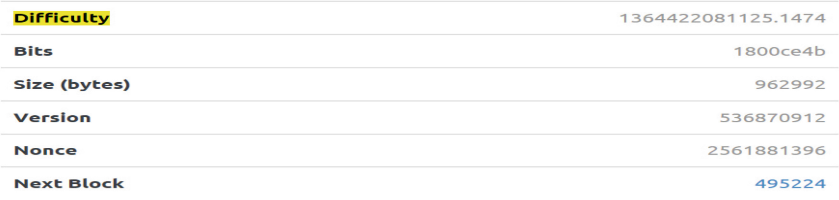
When running the script, we found the following results as depicted in Fig. 1. The calculated difficulty matches with the difficulty displayed in the Block #495223 Information (see Fig. 2).

The difficulty is tightly linked to the hashing rate. When the hashing rate increases, the proof-of-work is found quickly and therefore the difficulty increases too to keep the proof-of-work finding around 10-min in average. Also, when proof-of-work discovery time is slower, the difficulty decreases. Table 1 illustrates the strong correlation between the difficulty and the hashing rate. It shows also that difficulty and the hash



```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/lam/Desktop/Conferences/EMENA-ISTL 2018/script1.py ====
Max target Value in Dec(Block#0)= 26959535291011309493156476344723991336010898738574164086137773096960
Max Taget in Hex=0xffff0000000000000000000000000000000000000000000000000000
Current target Value in Dec(Block#495223)= 19758940920085072387393228723348383373068660102939017216
Current Taget in Hex(Block#495223)=0xce4b000000000000000000000000000000000000000000
Difficulty= 1364422081125.15
```

**Fig. 1.** Difficulty calculation of Block #495223.



| | |
|---|---|
| Difficulty | 1364422081125.1474 |
| Bits | 1800ce4b |
| Size (bytes) | 962992 |
| Version | 536870912 |
| Nonce | 2561881396 |
| Next Block | 495224 |

**Fig. 2.** Block #495223 Information [9].

**Table 1.** Difficulty and hash rate change between 2016 and 2017 [10]

| Date | Difficulty | Hash rate (GH/s) |
|---|---|---|
| Dec 6th, 2017 | 1,590,896,927,258 | 11,388,083,790 |
| Dec 2nd, 2016 | 286,765,766,821 | 2,052,749,317 |
| Ratio of change | 5.547722606133257 | 5.547722605818799 |

rates have quintupled since the last year. This is due mainly to competition between miners.

Without the difficulty, any miner possessing big hashing power would take over the Blockchain and could change it at will, therefore the difficulty participates strongly to the security of the Bitcoin.

## 4 Conclusion

All the aforementioned concepts suggest that Bitcoin is a secure by design crypto-currency. Its security relies on the cutting-edge cryptographic technologies such as the digital signature and the hash functions. The Difficulty plays a major role in the Bitcoin Security since it regulates the mining process, so a new block is added to the Blockchain within 10 min in average. Also, its dynamic change helps keep up with the increasing hashing rate to avoid Blockchain hijacking by miners with huge computational power. Notwithstanding the difficulty benefits, there is a big issue that Bitcoin community should address, which is the huge electricity consumed by the Miners using their hashing machines to overcome the difficulty.

Finally, to preserve the Bitcoin security, the community should empower the proof-of-work concept while searching for other computing alternatives so that the hashing process would become more energy efficient.

## References

1. Coin market capitalization. https://coinmarketcap.com/currencies/bitcoin/#charts. Accessed 26 Dec 2017
2. Certicom Research. Standards for Efficient Cryptography. SEC 2: Recommended Elliptic Curve Domain Parameters. (n.d.). http://www.secg.org/sec2-v2.pdf. Accessed 26 Dec 2017
3. Daulay, R.S.A., et al.: IOP Conference Series: Materials Science and Engineering, vol. 260, p. 012002 (2017)
4. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol with chains of variable difficulty. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017. Lecture Notes in Computer Science, vol. 10401. Springer, Cham (2017)
5. Eyal, I., Gencer, A.E., Sirer, E.G., van Renesse, R.: Cornell University. Bitcoin-NG: A Scalable Blockchain Protocol. https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf. Accessed 10 Aug 2018
6. Bitcoin Difficulty. https://en.bitcoin.it/wiki/Difficulty. Accessed 26 Dec 2017
7. Block Explorer. https://blockexplorer.com/block/00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048. Accessed 27 Dec 2017
8. Andreas, M.: Antonopoulos. Mastering Bitcoin: Programming the Open Blockchain, 2nd edn. (2017)
9. Block Explorer. https://blockexplorer.com/block/0000000000000000004e3c9d483093f88760b3c4c7083308785f6c880f81ab31. Accessed 10 Aug 2018
10. Bitcoin difficulty. https://bitcoinwisdom.com/bitcoin/difficulty. Accessed 27 Dec 2017