IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 8, August 2019

Proof-of-CAPTCHA: A True ASIC-Resistant Consensus

Bahaedinne Jlassi[1], Liuyang Ren[2], Scott Chen[3]

MASc, Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada[1]

PhD Candidate, Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada[2]

Professor, Electrical and Computer Engineering, Conestoga College, Cambridge, Canada[3]

Abstract: In *Proof-of-Work (PoW)* based blockchain systems, rewards received by miners are proportional to the computation power under their control, hence they are incentivized to use powerful hardware such as ASIC. This prevents commodity computers from joining the network and keeping the system decentralized, and leads to widespread waste of computation power and electrical energy for personal benefit. In this paper, we propose a new consensus algorithm named Proof-of-CAPTCHAs (*PoCA*), which employs CAPTCHAs upon completion of each block computation in order to achieve an ASIC-resistant nature while preserving user anonymity.

Keywords: Blockchain, Cryptocurrency, ASIC-Resistant, Proof-of-CAPTCHA

I. INTRODUCTION

The main distinguishing feature of permissionless blockchain systems is that they provide open participation while ensuring resistance against Sybil attack. This feature has never been achieved with traditional Byzantine fault tolerant protocols, such as PBFT [1]. Many cryptocurrencies, such as Bitcoin, use Proof-of-Work (*PoW*) to achieve the above properties. However, in the past decade, the market has seen miners periodically upgrade mining rigs to compete with each other for the appealing profits yielded from their mining efforts. Eventually, individuals who cannot afford high performance mining hardware are not incentivized to remain as miners. To make matters worse, the overall trend of muscling up mining rigs for profit generation creates a direct impact to the environmental sustainability; miners who deploy powerful hardware for personal benefits have developed an irresponsible attitude towards controlling energy consumption. As of today, no means to stop such wasteful behaviour have been seen. Bitcoin mining has gone through several eras: *CPU, GPU, FPGA, and ASIC* [2]. Nowadays, Bitcoin mining difficulty has spiked up to a prohibitive level, making it almost impossible for a new miner with a commodity computer to enter the mining workforce. Moreover, certain mining pools cartels and exchanges have gained significant influence over the market by amassing large number of Bitcoins, thereby creating a new form of “chaotic central banks”. It is a direct threat to the blockchain’s basic premise: decentralization. [3,4,5,6,7]

The proof-of-individuality (PoI) project creates anti-Sybil-tokens by relying on video pseudonym parties and Ethereum smart contracts [17]. The main security disadvantage is videos may show a recording instead of an actual livestream. Borge et al propose Proof-of-Personhood (PoP) to provide not only resistance against Sybil attacks, but also a fair wealth creation process [18]. Their work involves massive cryptographic concepts and rely on a Byzantine consensus protocol created by themselves, therefore, PoP is a complex protocol and difficult to understand, analyse, and ultimately deployed in practical uses. We propose Proof-of-CAPTCHA (*PoCA*) to restrain the advantage of powerful mining hardware, help stabilize the crypto-network, discouraging irresponsible energy consumption, and incentivizing new miners to join. As far as we know, cryptographic CAPTCHAs is currently the only possible way to ensure ASIC-resistance.

II. BACKGROUND

A. Cryptographic CAPTCHA

CAPTCHAs is the acronym of Completely Automated Public Turing test to tell Computers and Humans Apart. It is a type of challenge–response test used in computing to determine whether the user is human. As far as we know, only few people were interested in CAPTCHAs as a tool for achieving general cryptographic tasks. The most developed would be the work done by Abishek Kumarasubramanian [9], who deployed CAPTCHAs for a straight-line extractable commitment scheme and demonstrated how to incorporate it into the framework of zero-knowledge and UC secure protocols. Von Ahn, Blum, Hopper and Langford utilized CAPTCHAs for image-based steganography [10]. Canetti, Halevi and Steiner construct a scheme around CAPTCHAs to prevent off-line dictionary attacks on encrypted data [11].

DOI 10.17148/IJARCCE.2019.8803

Sandra and Debrup present an encryption protocol using CAPTCHAs that is secure against non-human profiling adversaries [12]. And finally, Dziembowski constructs a “human" key agreement protocol using only CAPTCHAs [13].

B. ASIC-Resistance

One of the most abused and misused cryptocurrency notion in recent years is once-concerning claim of ASIC-resistance, a term mostly used as an advertisement gimmick rather than a truthful presentation of cryptocurrency capabilities. In general, ASIC can be designed and manufactured to perform all existent hash functions [14]. In the crypto world, often a new hype is forged around a “fake” ASIC resistant algorithm. For instance, a big ASIC-resistance fuzz has recently been developed around the new Lyrar2v2 algorithm. However, upon closer investigation, the Vertcoin Lyra2v2 turns out to be simply ASIC non-friendly; it would merely fork to animate such resistance, referred to as an AAHF (Anti-Asic Hard Fork) [15].

III. INCREMENTAL BLOCK REWARD (*IBR*)

Proof-of-CAPTCHA (*PoCA*) is a two-layer consensus protocol that theoretically could be built on top of any existing blockchain consensus algorithms. The top level is restricted to a challenge-response type that uses cryptographic CAPTCHA. Because CAPTCHAs cannot be solved by machines, including ASICs, the presence of a human in front of a computer is inevitable. Solving a CAPTCHA challenge proves that a human is involved in the mining process without disclosing any information about this person, hence the feature "user anonymity". The nature of the bottom level depends on the second consensus algorithm *X* in use (e.g. *PoW*). As an illustration of the concept, we chose a Bitcoin-like *PoW* protocol as an example to be in the second level in fig.1.

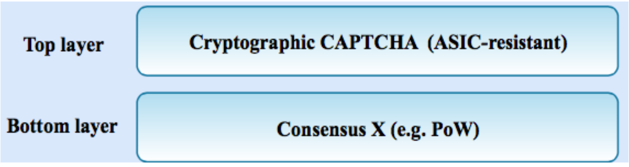
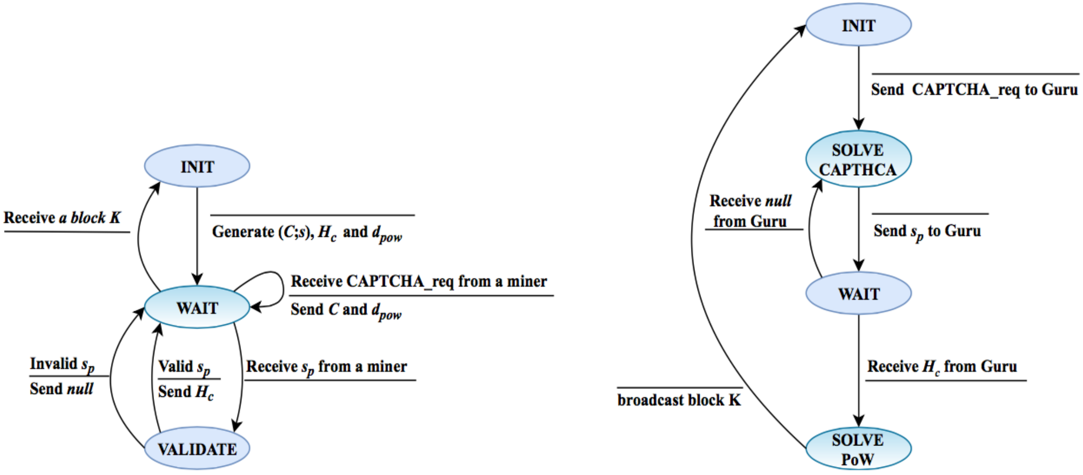


Fig.1 : *PoCA Structure*

A. Network Model

To ensure the implementation of *PoCA*, we incorporate in the network different roles and functionalities for the nodes. Guru Nodes, in short *GN*, generate and verify the CAPTCHA-solution pair *(C;s)*. They are full nodes that can autonomously and authoritatively verify any transaction without external reference. Other nodes can ask updates about the network and they need to be aware of the *GN* list. *GN* detail functionalities, qualification conditions, reward programs, and other relevant features are still under development. Like Bitcoin’s block header, a block header under *PoCA* also contains block metadata: version, previous block hash, merkle root, timestamp, difficulty target, and nonce. We add another field-*CAPTCHAstamp*-the hash function created for the pair *(C;s)*.

B. PoCA Algorithm



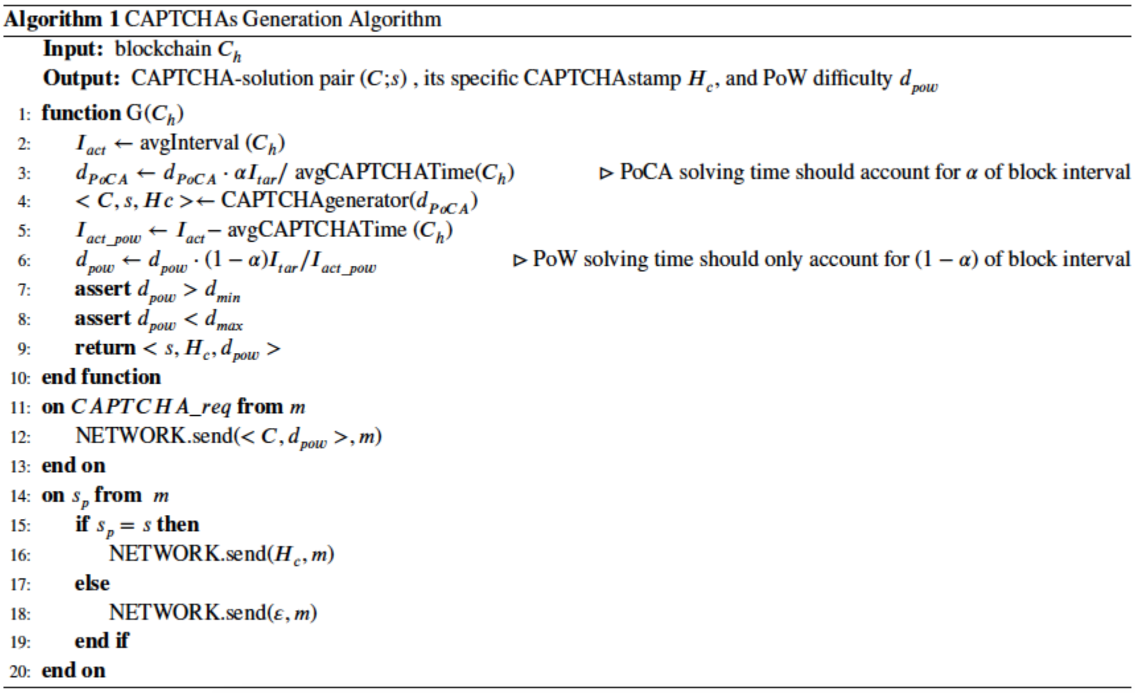
(a) Guru node **(b)** Miners

Fig.2 : *State Machine Diagrams*

CAPTCHA puzzles are the key in distinguishing human from a machine or ASIC, hence the true ASIC-resistant nature of our proposed method. The CAPTCHA puzzles are modelled using a modified notation proposed by Abishek [9].

DOI 10.17148/IJARCCE.2019.8803

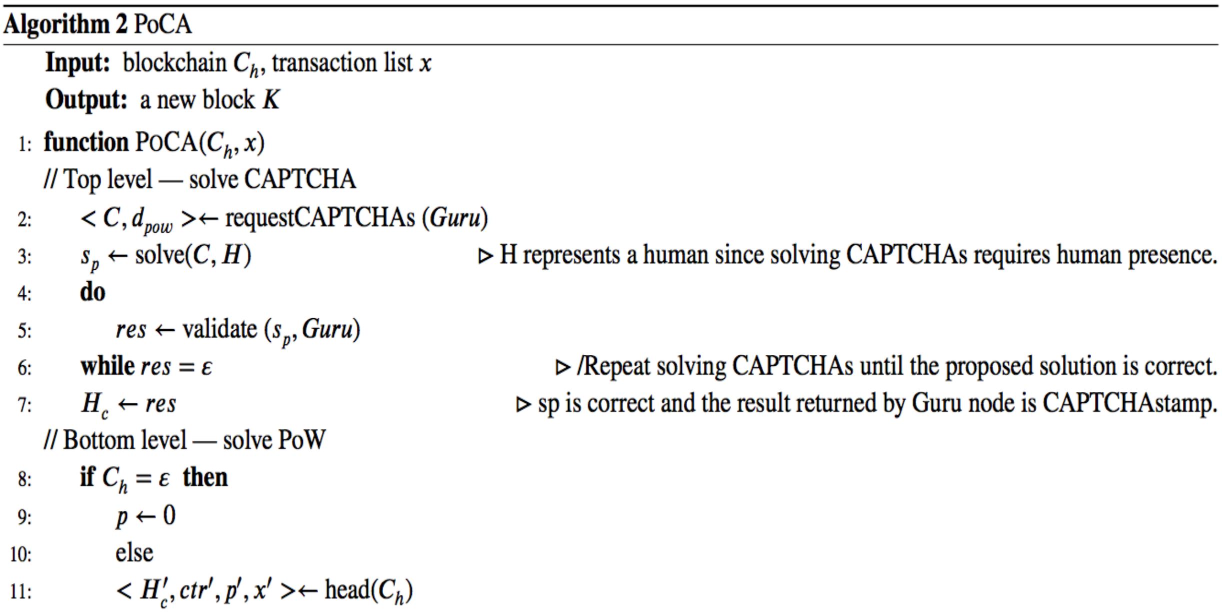
Fig.2a and Fig.2b show the state transitions of a *GN* and a mining node respectively. This mechanism allows us to distinguish between humans and machines.
A human *H* is capable of solving CAPTCHA puzzles. *GN*s generate a cryptographic CAPTCHA puzzle by running a generation algorithm denoted by *G* in fig.3. This algorithm outputs a CAPTCHA-solution pair (C;s) and its specific *CAPTCHAstamp* . Block generation under *PoCA* has two phases. A miner contacts a *GN* to get the CAPTCHA challenge *C* and *PoW* difficulty d_{pow}. Once a solution is found, the miner sends the proposed solution *sp* to the *GN*. The *GN* verifies whether *sp* is correct. If so, the *GN* sends H_c back to the miner; If not, it sends an empty string *ε*. The miner keeps solving
C and sending s_p to *GN* until H_c is received, and then the miner can proceed to assemble a block with collected transactions and solve *PoW*.



```
Algorithm 1 CAPTCHAs Generation Algorithm
  Input: blockchain $C_h$
  Output: CAPTCHA-solution pair $(C;s)$ , its specific CAPTCHAstamp $H_c$, and PoW difficulty $d_{pow}$
function G($C_h$)
    $I_{act} \leftarrow$ avgInterval ($C_h$)
    $d_{PoCA} \leftarrow d_{PoCA} \cdot \alpha I_{tar} /$ avgCAPTCHATime($C_h$)      ⊳ PoCA solving time should account for $\alpha$ of block interval
    $< C, s, Hc > \leftarrow$ CAPTCHAgenerator($d_{PoCA}$)
    $I_{act\_pow} \leftarrow I_{act} -$ avgCAPTCHATime ($C_h$)
    $d_{pow} \leftarrow d_{pow} \cdot (1-\alpha) I_{tar} / I_{act\_pow}$      ⊳ PoW solving time should only account for $(1-\alpha)$ of block interval
    assert $d_{pow} > d_{min}$
    assert $d_{pow} < d_{max}$
    return $< s, H_c, d_{pow} >$
end function
on $CAPTCHA\_req$ from $m$
    NETWORK.send($< C, d_{pow} >, m$)
end on
on $s_p$ from $m$
    if $s_p = s$ then
        NETWORK.send($H_c, m$)
    else
        NETWORK.send($\varepsilon, m$)
    end if
end on
```

Fig.3 : *CAPTCHA generation Algorithm*

Once succeeding in solving the *PoW*, the miner broadcast the newly generated block *K* to the network. The whole mining process is summarized in fig.4.



```
Algorithm 2 PoCA
  Input: blockchain $C_h$, transaction list $x$
  Output: a new block $K$
function PoCA($C_h, x$)
  // Top level — solve CAPTCHA
    $< C, d_{pow} > \leftarrow$ requestCAPTCHAs ($Guru$)
    $s_p \leftarrow$ solve($C, H$)      ⊳ H represents a human since solving CAPTCHAs requires human presence.
    do
        $res \leftarrow$ validate ($s_p, Guru$)
    while $res = \varepsilon$      ⊳ /Repeat solving CAPTCHAs until the proposed solution is correct.
    $H_c \leftarrow res$      ⊳ sp is correct and the result returned by Guru node is CAPTCHAstamp.
  // Bottom level — solve PoW
    if $C_h = \varepsilon$ then
        $p \leftarrow 0$
        else
        $< H'_c, ctr', p', x' > \leftarrow$ head($C_h$)
```

```
12:         $p \leftarrow \mathrm{H}\,(H_c', ctr', p', \mathrm{M}(x'))$
13:     end if
14:     $ctr \leftarrow 1$
15:     $K \leftarrow \varepsilon$
16:     $h \leftarrow \mathrm{M}(x)$
17:     $T \leftarrow T_{max}/d_{pow}$
18:     while $ctr \leq q$ do
19:         if $\mathrm{H}\,(Hc, ctr, p, h) < T$ then
20:             $K \leftarrow < H_c, ctr, p, x >$
21:             break
22:         end if
23:         $ctr \leftarrow ctr + 1$
24:     end while
25:     return $K$
26: end function
```

Fig.4 : *PoCA Algorithm*

In fig.3 and fig.4 , *avgInterval*(C_h)*(.)* calculates the average block interval over the last 2016 blocks (same as Bitcoin) I_{act} is the actual average block interval; I_{tar} is the target block interval; d_{PoCA} is the CAPTCHA's difficulty; *avg*CAPTCHATime(C_h)(.) calculates the average CAPTCHA solving time over the last, for example, 2016 blocks; I_{act_pow} is the actual average *PoW* solving time. d_{pow} is the *PoW* difficulty.

The CAPTCHA layer itself is ASIC-resistant but vulnerable to Sybil attack. A miner can run multiple mining software instances and reproduce the solution on other instances once the CAPTCHA challenge is solved on one instance. In contrast, the *PoW* layer is vulnerable to ASIC mining but Sybil-attack-resistant. To strike a balance between the two, d_{pow} must be tuned carefully. Upper bound and lower bound are set for this purpose: d_{min} is the lower bound to prevent Sybil attack; d_{max} is the upper bound to ensure little benefit of using *ASIC*.

Guru is the network address info of a *GN*; *res* is the s_p validation result returned by the *GN*; ε is the empty string. CAPTCHA*req* is the request from a miner asking for the CAPTCHA challenge for the next block; *m* is a miner; *x* is a list of transactions to be included in the next block; C_h is the current blockchain from the miner's view; *ctr* is the nonce value; *p* is the header hash of the previous block; *head(.)* returns latest block on a blockchain. The latest block is represented by $<H_c´, ctr´, p´, x´>$ [16];*T* is the threshold for a block header hash to be valid; T_{max} is the maximum target hash; *H(.)* is a hash function, e.g. SHA256. *M(.)* is the Merkle tree function*;* *h* is the Merkle root; *q* is the number of times the algorithm is allowed to brute-force the hash function inequality in line 19.

C. *PoCA Analysis*

Assume miner m controls a single processing power unit, i.e. m can compute 1 hash per second. Every hash has the probability of $1/d_{pow}$ to be valid. If m solves *PoW* at t^{th} second, it must be the case that all the hashes before (t-1)th second are invalid and the one at t^{th} second is valid. Thus, single processing power unit *PoW* solving time conforms to geometric distribution with success probability $1/d_{pow}$. According to the property of geometric distribution, the mean value of single processing power unit *PoW* solving time is d_{pow}, the reciprocal of success probability. Similarly, for miners holding *n* processing power units, the mean *PoW* solving time is d_{pow}/n.

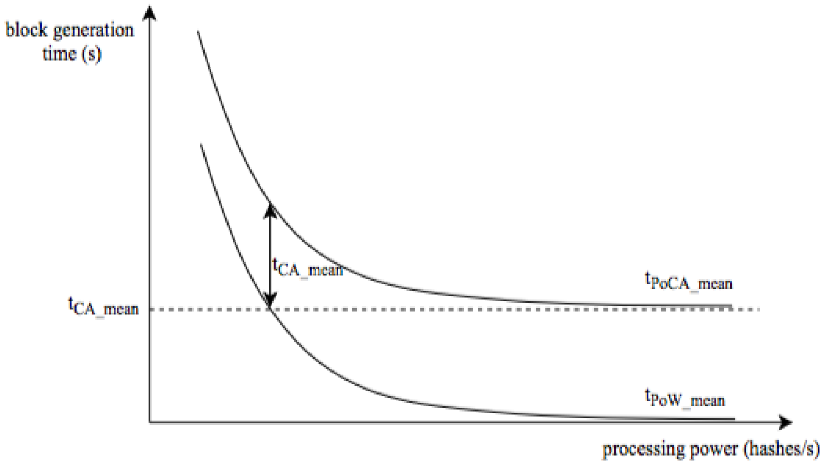


Fig.5 Mean block generation time against processing power units under PoCA

CAPTCHA must be solved by a human, so we can safely assume CAPTCHA solving time conforms to a normal distribution, and its mean value is denoted by t_{CA_mean}. Since a miner must solve both CAPTCHA and *PoW*, the mean block generation time against processing power units owned by a miner is plotted in fig.5. t_{CA_mean} becomes increasingly dominant as the processing power increases, thereby effectively restraining the advantages of large miners, and ultimately controls the amount of electrical energy consumed by the *PoCA*-manifested cryptocurrency network.

IV. PoCA SIMULATION

To demonstrate the ASIC-resistant property of *PoCA*, we have executed a Monte-Carlo simulation. A random function is employed to choose a lucky miner who has the privilege to propose the next block and be rewarded. The network is configured as 100 miners where 60, 30 and 10 miners control 1 unit, 1.5 units and 2 units hash power respectively to simulate the non-uniform distribution of computing power. The coin supply is 1.68 million in total. fig.6 illustrates that with *PoW*, wealth is far from evenly distributed. Miners with higher computation capacity, i.e. more powerful mining hardware like ASIC, gain more coins. Specifically, the expected rewards of miners with 1 unit, 1.5 units and 2 units hash power are 13440 coins, 20160 coins and 26880 coins respectively.

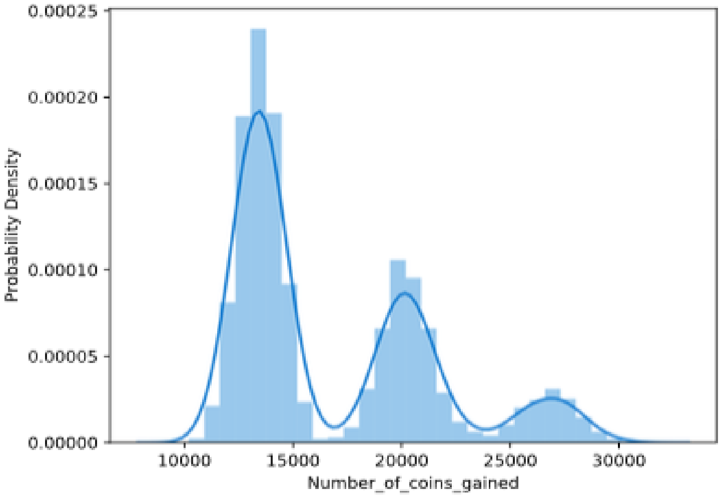


Fig.6 Wealth distribution in PoW case

CAPTCHA solving time is set as 99% of the total *PoCA* solving time. The simulation results in fig.7 shows *PoCA* significantly restrains the advantage of high-end mining hardware and thus even out the wealth distribution.

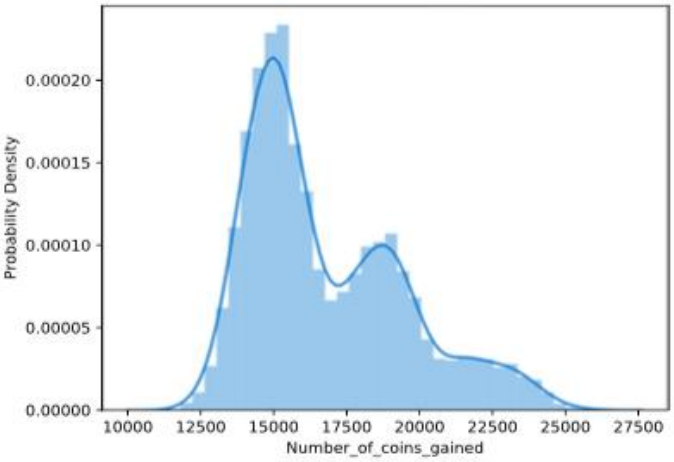


Fig.7 Wealth distribution in PoCA case

V. CONCLUSION

We proposed *PoCA* in this paper and demonstrated its resistance against ASIC. *PoCA*-based blockchain systems is designed to help consume less energy, encourages new miners to join, and enables fair wealth creation.

DOI 10.17148/IJARCCE.2019.8803

REFERENCES

[1]. Castro M, Liskov B, others . Practical Byzantine fault tolerance. In: . 99. USENIX. ; 1999: 173–186.
[2]. Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press . 2016.
[3]. Andrew M. Provable security for cryptocurrencies. Doctoral dissertation. University of Maryland, Maryland; 2016.
[4]. Muneeb A. Trust-to-trust design of a new Internet. Doctoral dissertation. Princeton University, New Jersey; 2017.
[5]. Adem Efe G. On Scalability of Blockchain Technologies. Doctoral dissertation. Cornell University, New York; 2017.
[6]. N. A. Blockchain Consensus and Beyond: Scalable Secure Consensus & Applications. Project proposal; 2017.
[7]. SnehaG.Scalabilityanalysisofblockchainsthroughblockchainsimulation.masterthesis.UniversityofNevada.LasVegas: 2017.
[8]. König T, Duran E. FairCoin V2 white paper. Faircoin.org 2016.
[9]. AbishekK.,ed.,Cryptographyusingcaptchapuzzles (Berlin,Heidelberg); Public-KeyCryptography–PKC2013,Springer:2013. pp. 89-106.
[10]. Von Ahn L, Blum M, Hopper NJ, Langford J. CAPTCHA: Using hard AI problems for security. In: Springer. ; 2003: 294–311.
[11]. Canetti R, Halevi S, Steiner M. Mitigating dictionary attacks on password-protected local storage. In: Springer. ; 2006: 160–179.
[12]. Dıaz-Santiago S, Chakraborty D. On securing communication from profilers. 2012.
[13]. Abishek K. Connecting Theory and Practice in Modern Cryptography. Doctoral dissertation. UCLA, California; 2014.
[14]. Larry r. Proof of stake velocity: building the social Currency of the digital age. Self-published white paper 2014.
[15]. Kay K. kaykurokawa/forking-for-asic-resistance-a-monero-case-study-ecdfb6c9fba2 . 2017.
[16]. Juan Gea. , ed., The bitcoin backbone protocol: Analysis and applications(Berlin, Heidelberg); Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer: 2015. pp. 89-106.
[17]. Proof of individuality, newsbtc.com/2016/04/05/proof-of-individuality-blockchain-security Accessed: 2018-10-19.
[18]. Borge M, Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Ford B. Proof-of-Personhood: Redemocratizing Permission-less Crypto currencies. IEEE. ; 2017: 23–26.

BIOGRAPHY

Bahaedinne Jlassi is the CEO of Demystify. He is a Blockchain technology expert with a vision to popularize it around the world. Baha is an Electrical and Computer Engineer with two master degrees (MASc) from the University of Waterloo and the Superior Institute of Technology in Montreal (ETS). Baha has more than 10 years of experience in both industrial and academic fields. He worked as a lab instructor at the University of Waterloo for years teaching digital electronics, VHDL, FPGA, embedded system design and Physics. As an engineer, he has supervised and managed multiple projects in Electronics, Information Technology and Telecommunication. Baha is a hungry reader, an avid traveller, a fitness passionate, a debutant backpacker, an acceptable cook and an adventure seeker.

DOI 10.17148/IJARCCE.2019.8803