WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

Dynamic Cryptography Integrated Secured Decentralized Applications with Blockchain Programming

Jamal Khmadhloom[1,*]

[1] Computer Sciences and information Technology College, Wasit University, Iraq

*Corresponding Author: Jamal Kh-Madhloom

DOI: https://doi.org/10.31185/wjcm.Vol1.Iss2.41
Received: February 2022; Accepted: April 2022; Available online: June 2022

ABSTRACT: Blocks and chains are the building blocks of the blockchain, which is a community network. Blocks and chains are two terms used to describe collections of data information. The most fundamental need for a blockchain is that these postings be connected by cryptography, which is the case here. Cryptography. the entries in each block are added to as the list grows. Although the concept of blockchain cryptography is difficult, we have made it easier for you to understand. Asymmetric-key cryptography and hash functions are used in blockchains. Hash functions provide participants with a complete image of the internet. The SHA-256 hashing algorithm is often used in blockchains. In Bitcoin, where addresses are tracked by public-private key pairs, blockchains are often used. The public key in blockchain cryptography is a person's address. All participants have access to the participant's public key. The private key is used to get access to the address database and to authorise activities using the address. To ensure the integrity of the blockchain ledger, encryption plays a key role. Each event on the blockchain is recorded using encrypted data. As long as each user has access to their cryptographic keys, they may buy or trade cryptocurrencies. The root hashes of all transactions are stored in blockchains via cryptographic hashing. If somebody attempts to tamper with any data upon that blockchain, the main hash will have a completely new hash. Root hash comparisons may be performed on any other system to check whether the data is safe.

Keywords: Blockchain, Dynamic Cryptography, Security in Blockchain

1. INTRODUCTION

As a leading technology, blockchain is usually linked with high levels of security and anonymity across a wide range of applications. Blockchain technology is now being used in a wide range of social and business contexts, not only in the cryptocurrency industry. E-governance, social networking and e-commerce are just a few of the many sub-segments that fall under this umbrella [1].

Secured Digital Ledgers with Cryptography in Blockchain

Digital ledgers are stored on a high-security, high-performance system known as a "blockchain." Using a digital ledger eliminates the need for intermediaries or administrators to manipulate the records [2]. All operations on the bitcoin blockchain are finalised using a variety of protocols and processes that cannot be hacked by external parties. [3].

Key Implementations of Blockchain Technology

- **Entertainment**
  - Spotify
  - Guts

*Corresponding author:

  - B2Expand
  - KickCity
  - Veredictum
- **Social Networks**
  - Matchpool
  - MeWe
  - Minds
  - Steepshot
  - Mastodon
  - DTube
  - Sola
- **Retail**
  - Opskins
  - Loyyal
  - Warranteer
  - Every.Shop
  - Blockpoint
  - Fluz Fluz
  - Spl.yt
  - Shopin
  - Ecoinmerce.io
  - Portion
  - Buying.com
- **Cryptocurrency**
  - Litecoin
  - Ripple
  - Primecoin
  - Bitcoin
  - Namecoin
  - Dogecoin
  - Nxt
  - Ethereum

2. REAL PROBLEM AND KEY STATEMENT

In traditional centralized environment, there is less security because of weak hashing and needs dynamic hash with cryptography as in Blockchain. Transactions cannot be reversed because cryptographic hashing is irreversible. This assures that all users can rely on the digital ledger's correctness and that they are protected from any antagonistic conduct [4, 5].

To understand blockchain, one must understand cryptography. It is possible to encrypt, transmit bitcoin securely, and record transactions over time thanks to cryptography's features. Without a central authority, we may trade bitcoin safely and assure that blocks will continue to be inserted into the chain without restriction [6].

Cryptographic hashing enables blockchains to store vast quantities of transactions while protecting them from hackers. It provides a secure, verifiable, and scalable method for conducting online transactions. Blockchain is genuinely unstoppable because of cryptography [7, 8].

2.1 PUBLIC AND PRIVATE BLOCKCHAIN WITH SECURITY AND CRYPTOGRAPHY

The data structure of blockchain technology is automatically safe. Encryption and decentralisation are the foundations upon which it is based to ensure transactional trust. In most blockchains and DLT, each block of data contains a single transaction or a collection of related transactions. It is extremely impossible to tamper with a cryptographic chain since each new block is irrevocably tied to the blocks that came before it. To ensure that each transaction contained in a block is correct, each transaction must be verified and agreed upon by the consensus process [9, 10].

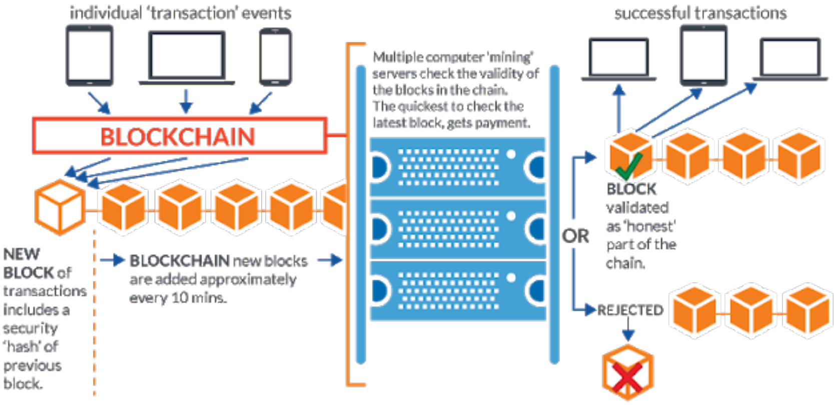

individual 'transaction' events
BLOCKCHAIN
NEW BLOCK of transactions includes a security 'hash' of previous block.
BLOCKCHAIN new blocks are added approximately every 10 mins.
Multiple computer 'mining' servers check the validity of the blocks in the chain. The quickest to check the latest block, gets payment.
OR
successful transactions
BLOCK validated as 'honest' part of the chain.
REJECTED

FIGURE 1. **Dynamic Cryptography Based Security in Blockchain**

Blockchain technology is able to transform society since it relies on the participation of all users in a network [11, 12]. There is no single point of failure because everything is documented. But when it comes to digital money security, blockchain technology has a few advantages over the alternatives. Participation and data access might differ amongst blockchain networks. Public and private networks are the two most common varieties.

A common feature of public blockchain networks is the ability for anybody to join while yet maintaining their anonymity. Using computers linked to the internet, a public blockchain can verify transactions [13] and establish consensus. Public blockchains, like as Bitcoin, use "bitcoin mining" as a means of bringing parties together. The "miners" on the bitcoin network work cooperatively to solve a complex cryptographic challenge in order to validate a transaction. Other than public keys, this type of network has few means of identity and access control.

In private blockchains, where only well-known companies are allowed to participate, identity is employed to authenticate membership and access credentials [14–16]. The organisations' combined efforts establish an exclusive "business network." Consensus on a private blockchain in a permissioned network is achieved by "selective endorsement."

Before choosing a network for any blockchain application, think about the requirements of your firm. It is better for requirements to use permissioned and private networks, as they may be more tightly controlled [17, 18]. Decentralization and dispersion can take place more easily in permissionless networks.

3. KEY OBJECTIVE AND RELATED ASPECTS

The unique verification process of Blockchain is one of its outstanding features. By removing the need for human verification, Blockchain promises to increase accuracy. Third-party verification costs have also been reduced as a result of the use of the blockchain technology [19]. Because of its decentralised design, hackers and attackers have had a tough time tampering with data. Security, privacy, and efficiency are just a few of the benefits of using Blockchain to do business. Since it is a transparent technology, it provides complete transparency to its users. As a result, inhabitants of countries with unsafe or undeveloped governments can use Blockchain as a financial option as well as a way to protect their personal information [20]. It's still early days for blockchain, but there are countless chances for professionals to study and grow their careers in this industry, including cryptography in blockchain for sure.

There are blocks of various data items and documents in the blockchain network. It is impossible to alter a block after it has been added to the blockchain. In this context, "immutable" refers to the fact that it cannot be changed or tampered with. Consequently, it creates a secure chain of blocks that eliminates the risk of data manipulation or leaking [21].

The Genesis Block is the initial entry in the chain of the network and is where the blockchain begins transactions. It

Table 1. **Key Tools for Secured Blockchain Environment**

URL	Framework / Tool
https://etherlime.readthedocs.io/en/latest/	Etherlime
https://ethfiddle.com/	EthFiddle
https://embark.status.im/	Embark
http://populus.readthedocs.io/en/latest/	Populus
http://remix.ethereum.org/	Remix IDE
https://truffleframework.com/	Truffle
https://geth.ethereum.org/ https://github.com/ethereum/go-ethereum/wiki/geth	Go Ethereum / Geth
https://consensys.net/diligence/mythril.html	MyThril
https://github.com/cryppadotta/dotta-license/tree/master/dot-abi-cli	Dot-Abi-cli
https://github.com/ethereum/pyethereum	PyEthereum
https://nethereum.com/	Nethereum
https://github.com/consensys/cava	Cava
http://www.liquidity-lang.org/	Liquidity
https://infura.io/	Infura
https://lamden.io/	Lamden
http://solidity.readthedocs.io/en/v0.4.24/	Solidity
https://coq.inria.fr/	Coq

gets more difficult to decipher the prior states because of the various encryptions that have been added to each new block that is inserted.

Table 2. **Literature Review**

Authors	Key Work
G. Zyskind, A.S. Pentland [22]	Using Secured Cryptography based blockchain, data exchanges between users and apps may be protected and undamaged. Network nodes reward trustworthy nodes for their degree of trust rather than proof-of-work.
B. Benshoof, A. Rosen, A.G. Bourgeois, R.W. Harrison [23]	Cryptography Hash with Dynamic Security and "D3NS" with blockchain-based solution for securing DNS. New DNS proposed that is backwards compatible.
A. Ouaddah, A. Abou Elkalam, A. Ait Ouahman [24]	PoC Based Implementation
M. Ali, et al. [25]	Dynamic Cryptography based application., "BlockStack" is a test project for immutable data naming and storage. Recognizing that the Namecoin blockchain does not provide the same level of security and trustworthiness as the Bitcoin blockchain.
K. Christidis, M. Devetsikiotis [26]	IoT devices with implementation patterns with the use of blockchain technology are examined in detail.

4. METHODOLOGY

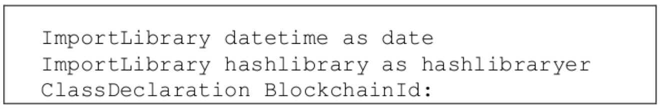
- Using Python based Programming Platform for Cryptography in Blockchain Development
- Development, Generation and Deployment of Dynamic Hash
- Using Dynamic Hash in Blockchain Environment

Python is the programming language of choice for high-performance computations in practically every field. The tools and frameworks provided by Python may be used to construct blockchain applications, including those that are decentralized.

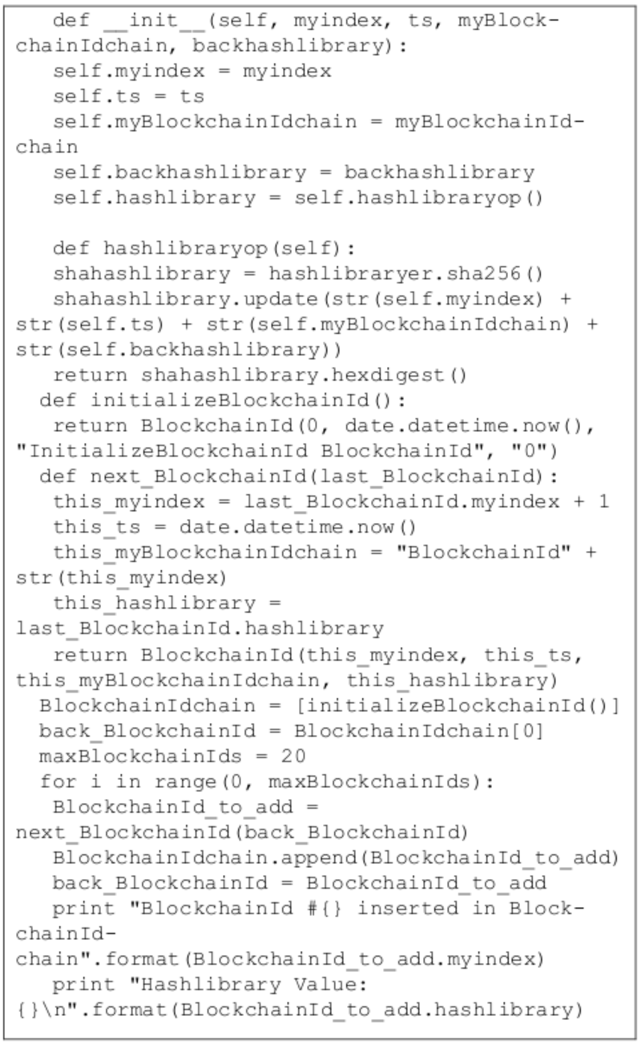
```
  $ pip install <packagename>

  MyDrive:\PythonInstallationDirectory>python -m pip in-
stall <packagename>
```

The incorporation of dynamic cryptographic and encryption into the blockchain technology is critical and heavily relied upon. Installing the hashlib library is as simple as following the instructions found in the previous paragraph.

To ensure that all transactions and records are safe, a secure blockchain generates hash values for each transaction and record. It is possible to produce the dynamic hash value needed to build a blockchain from a collection of individual transactions by running the following script, blockchainhash.py.



```
ImportLibrary datetime as date
ImportLibrary hashlibrary as hashlibraryer
ClassDeclaration BlockchainId:
```



```
   def __init__(self, myindex, ts, myBlock-
chainIdchain, backhashlibrary):
   self.myindex = myindex
   self.ts = ts
   self.myBlockchainIdchain = myBlockchainId-
chain
   self.backhashlibrary = backhashlibrary
   self.hashlibrary = self.hashlibraryop()

   def hashlibraryop(self):
   shahashlibrary = hashlibraryer.sha256()
   shahashlibrary.update(str(self.myindex) +
str(self.ts) + str(self.myBlockchainIdchain) +
str(self.backhashlibrary))
   return shahashlibrary.hexdigest()
  def initializeBlockchainId():
   return BlockchainId(0, date.datetime.now(),
"InitializeBlockchainId BlockchainId", "0")
  def next_BlockchainId(last_BlockchainId):
   this_myindex = last_BlockchainId.myindex + 1
   this_ts = date.datetime.now()
   this_myBlockchainIdchain = "BlockchainId" +
str(this_myindex)
   this_hashlibrary =
last_BlockchainId.hashlibrary
   return BlockchainId(this_myindex, this_ts,
this_myBlockchainIdchain, this_hashlibrary)
  BlockchainIdchain = [initializeBlockchainId()]
  back_BlockchainId = BlockchainIdchain[0]
  maxBlockchainIds = 20
  for i in range(0, maxBlockchainIds):
   BlockchainId_to_add =
next_BlockchainId(back_BlockchainId)
   BlockchainIdchain.append(BlockchainId_to_add)
   back_BlockchainId = BlockchainId_to_add
   print "BlockchainId #{} inserted in Block-
chainId-
chain".format(BlockchainId_to_add.myindex)
   print "Hashlibrary Value:
{}\n".format(BlockchainId_to_add.hashlibrary)
```

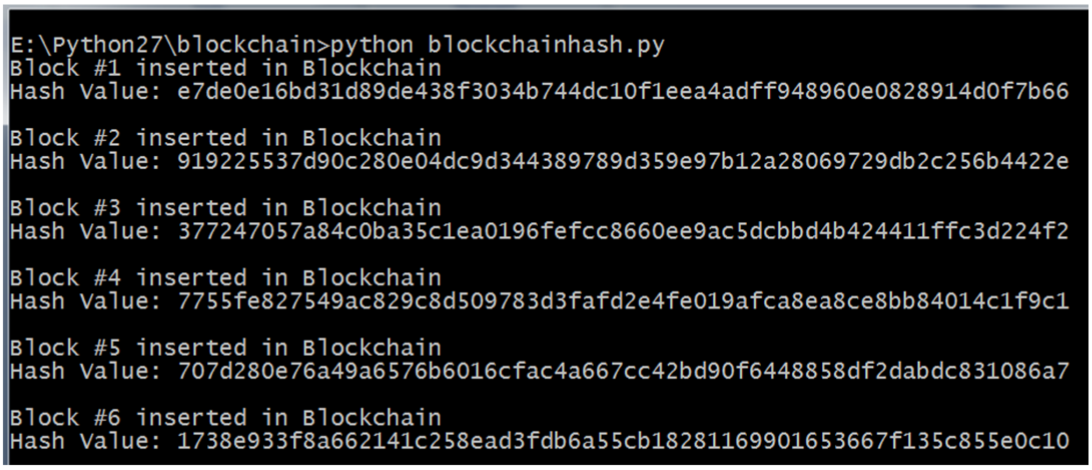
5. RESULT

The result of executing code is a different hash value and a better degree of security employing cryptography techniques. Attempts to hack or sniff the transaction will be nearly impossible if these hash values are used.

6. DEPLOYMENT OF NETWORK BASED DISTRIBUTED BLOCKCHAINS

Block-based hash functions, like those in preceding examples, are implemented on a standalone system. The real blockchain must be dispersed in order for various users to start their own transactions and blocks. Python has a variety of frameworks for distributed and web-based solutions.

Thus, the digital currency or transaction is carried out securely. B’s records need to reflect the value of a file or digital currency sent by A, for example, if A’s records need to be wiped and mirrored in B’s records. Traditionally, the Bank

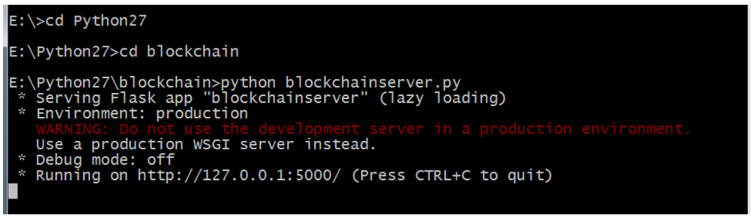


```
E:\Python27\blockchain>python blockchainhash.py
Block #1 inserted in Blockchain
Hash Value: e7de0e16bd31d89de438f3034b744dc10f1eea4adff948960e0828914d0f7b66

Block #2 inserted in Blockchain
Hash Value: 919225537d90c280e04dc9d344389789d359e97b12a28069729db2c256b4422e

Block #3 inserted in Blockchain
Hash Value: 377247057a84c0ba35c1ea0196fefcc8660ee9ac5dcbbd4b424411ffc3d224f2

Block #4 inserted in Blockchain
Hash Value: 7755fe827549ac829c8d509783d3fafd2e4fe019afca8ea8ce8bb84014c1f9c1

Block #5 inserted in Blockchain
Hash Value: 707d280e76a49a6576b6016cfac4a667cc42bd90f6448858df2dabdc831086a7

Block #6 inserted in Blockchain
Hash Value: 1738e933f8a662141c258ead3fdb6a55cb1828116990165366 7f135c855e0c10
```

FIGURE 2. **Secured Hash Generationin Blockchain**

has acted as a go-between in this transaction. On the blockchain, transactions are validated in real time by specialised algorithms, cutting out the middlemen. Currency will depreciate regardless of the kind of currency if the sender does not erase the transaction from their account.

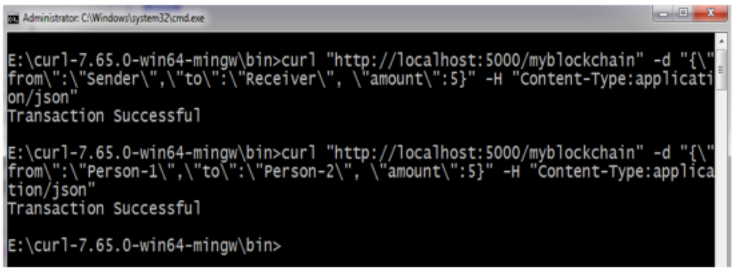
```
    CryptoMiner_address = "***************************"
    mySecuredBlockchain = []
    mySecuredBlock-
   chain.append(Creation_genesis_SecuredBlock())
    ThisClass_Secured_Nodes_SecuredTransactions = []
    peer_Secured_Nodes = []
    SecuredMining = True


  @Secured_Node.route('/mySecuredBlockchain', meth-
ods=['POST'])
  def SecuredTransaction():
   new_mySecuredBlockchain = request.get_json()
   This-
Class_Secured_Nodes_SecuredTransactions.append(new_mySecu
redBlockchain)
   Display "New SecuredTransaction"
   Display "Sender:
{}".format(new_mySecuredBlockchain['from'].encode('ascii'
,'replace'))
   Display "Receiver:
{}".format(new_mySecuredBlockchain['to'].encode('ascii','
replace'))
   Display "Amount:
{}\n".format(new_mySecuredBlockchain['amount'])
   return "SecuredTransaction Successful\n"
  @Secured_Node.route('/SecuredBlocks', methods=['GET'])
  def get_SecuredBlocks():
   chain_to_send = mySecuredBlockchain
   for i in range(len(chain_to_send)):
   SecuredBlock = chain_to_send[i]
   SecuredBlock_idx = str(SecuredBlock.idx)
   SecuredBlock_timestamp = str(SecuredBlock.timestamp)
   SecuredBlock_data = str(SecuredBlock.data)
   SecuredBlock_hash = SecuredBlock.hash
   chain_to_send[i] = {
   "idx": SecuredBlock_idx,
   "timestamp": SecuredBlock_timestamp,
   "data": SecuredBlock_data,
   "hash": SecuredBlock_hash
   }
```



```
E:\>cd Python27

E:\Python27>cd blockchain

E:\Python27\blockchain>python blockchainserver.py
 * Serving Flask app "blockchainserver" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

FIGURE 3. **Blockchain Initialization**

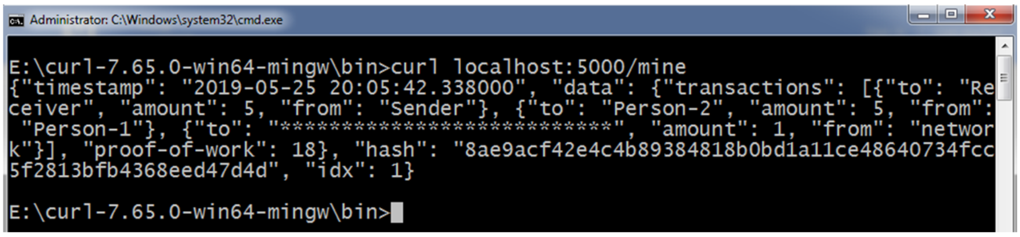
```
  $ curl "http://localhost:5000/blockchain" -d
"{\"from\":\"ss\",\"to\":\"fsd\", \"amount\":3}" -H "Con-
tent-Type:application/json"
```



```
E:\curl-7.65.0-win64-mingw\bin>curl "http://localhost:5000/myblockchain" -d "{\"
from\":\"Sender\",\"to\":\"Receiver\", \"amount\":5}" -H "Content-Type:applicati
on/json"
Transaction Successful

E:\curl-7.65.0-win64-mingw\bin>curl "http://localhost:5000/myblockchain" -d "{\"
from\":\"Person-1\",\"to\":\"Person-2\", \"amount\":5}" -H "Content-Type:applica
tion/json"
Transaction Successful

E:\curl-7.65.0-win64-mingw\bin>
```

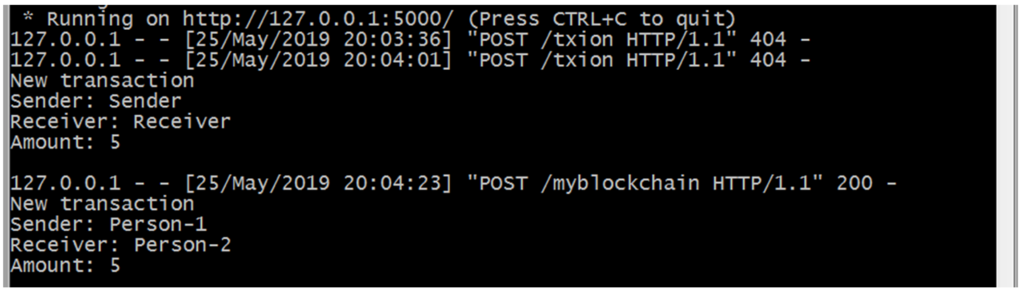
FIGURE 4. **Curl based Cryptography Hash Initialization**

Proof-of-work (PoW) is a critical algorithm in blockchain programming. It is used to verify and confirm transactions in order to add new blocks to the blockchain. The key consensus algorithm for verifying and authenticating transactions is referred to as such. In a blockchain network, there are a variety of miners that work together to verify and finalise transactions. The miners are compensated with digital crypto-currencies as compensation for their successful validations.



```
E:\curl-7.65.0-win64-mingw\bin>curl localhost:5000/mine
{"timestamp": "2019-05-25 20:05:42.338000", "data": {"transactions": [{"to": "Re
ceiver", "amount": 5, "from": "Sender"}, {"to": "Person-2", "amount": 5, "from":
 "Person-1"}, {"to": "***************************", "amount": 1, "from": "networ
k"}], "proof-of-work": 18}, "hash": "8ae9acf42e4c4b89384818b0bd1a11ce48640734fcc
5f2813bfb4368eed47d4d", "idx": 1}

E:\curl-7.65.0-win64-mingw\bin>
```

FIGURE 5. **Secured Transactions**



```
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [25/May/2019 20:03:36] "POST /txion HTTP/1.1" 404 -
127.0.0.1 - - [25/May/2019 20:04:01] "POST /txion HTTP/1.1" 404 -
New transaction
Sender: Sender
Receiver: Receiver
Amount: 5

127.0.0.1 - - [25/May/2019 20:04:23] "POST /myblockchain HTTP/1.1" 200 -
New transaction
Sender: Person-1
Receiver: Person-2
Amount: 5
```

FIGURE 6. **Secured Transaction with Cryptography in Users**

With the code execution and overall implementation described, there will be no effort at hacking because all data and transactions can be inspected to ensure complete transparency. It is possible to record and commit the integrity of transactions by utilising Proof of Work (PoW).

7. CONCLUSION

Currently, governments and corporations alike are working to protect their applications by implementing blockchain technology. Secure Proof of Work (PoW) algorithms must be linked to these integrations in order to ensure implementation privacy and integrity. Blockchain technology may be utilised by researchers and forensic scientists to accurately anticipate the identities of certain individuals, which can be employed in criminal forensic and law enforcement settings. Software (Electrum, Bitcoin core) and perhaps a specific hardware device (e.g. Ledger) can be used to store transaction data and the user's private and public keys (e.g. private/public key pair). It's critical to understand that these wallets do not hold any actual money (e.g. Bitcoin, Ethereum). There is nothing more to these wallets than a location to store one's private keys and transaction balance. A blockchain wallet is also required to conduct transactions with these other users. To put it another way, the blockchain holds all of the true information/data and cash in blocks, not a wallet. It's akin to a digital signature, which serves as a kind of identification for both the recipient and the whole blockchain network. A particular method must be used to combine your data and your cryptographic signature to establish a unique digital signature each time you begin a transaction with another node. As a result, you may rest assured that both your node and also the data it transmits are genuine.

FUNDING

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] T. Dvir, L. Holczer, and Buttyan, "VeRA - Version number and rank authentication in RPL," in *Proceedings of the 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 2709–2723, 2011.

[2] F. Álonso, L. Fernández, J. Marco, and Salvachúa, "IAACaaS: IoT Application-Scoped Access Control as a Service," *Futur Internet*, vol. 9, no. 4, pp. 64–64, 2017.

[3] H. Liu, B. Yang, and T. Liu, "Efficient Naming, Addressing and Profile Services in Internet-Of-Things Sensory Environments," *Ad Hoc Networks*, vol. 18, pp. 85–101, 2014.

[4] T. Conzon, P. Bolognesi, A. Brizzi, R. Lotito, M. A. Tomasi, and Spirito, "The Virtus Middleware: An Xmpp Based Architecture for Secure Iot Communications," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012.

[5] G. Lally and D. Sgandurra, "Towards a framework for testing the security of IoT devices consistently," in *Proceedings of the First International Workshop on ETAA 2018*, 2018.

[6] J. Granjal, E. Monteiro, and J. S. Silva, "Network-layer security for the internet of things using TinyOS and BLIP," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1938–1963, 2014.

[7] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," *Proceedings of the ACM Proceedings - ACM Conference on Computer and Communications Security*, pp. 254–69, 2016.

[8] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[9] M. Adil, M. A. Almaiah, A. O. Alsayed, and O. Almomani, "An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks," *Sensors*, vol. 20, pp. 1–19, 2020.

[10] M. Adil, M. A. Almaiah, A. O. Alsayed, and O. Almomani, "An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks," *Sensors*, vol. 20, no. 8, pp. 2311–2311, 2020.

[11] M. H. Ibrahim, "Octopus: an edge-fog mutual authentication scheme," *International Journal on Network Security*, vol. 18, pp. 1089–1101, 2016.

[12] M. Jamshidi, E. Zangeneh, M. Esnaashari, A. M. Darwesh, and M. R. Meybodi, "A novel model of sybil attack in cluster-based wireless sensor networks and propose a distributed algorithm to defend it," *Wireless Personal Communications*, vol. 105, no. 1, pp. 145–173, 2019.

[13] N. Park, "Mutual authentication scheme in secure internet of things technology for comfortable lifestyle," *Sensors (Switzerland)*, vol. 16, pp. 1–16, 2015.

[14] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity authentication and capability based access control (iacac) for the internet of things," *J. Cyber Secur. Mobil*, vol. 1, pp. 309–348, 2013.

[15] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes 2018," in *Proceedings of the 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, 2018.

[16] R. Riaz, K. H. Kim, and H. F. Ahmed, “Security Analysis Survey and Framework Design for Ip Connected Lowpans,” in *Proceedings of the 2009 International Symposium on Autonomous Decentralized Systems (IEEE)*, pp. 1–6, 2009.

[17] S. Dong, X. Zhang, and W. G. Zhou, “A security localization algorithm based on DV-hop against Sybil attack in wireless sensor networks,” *Journal of Electrical Engineering & Technology*, vol. 15, no. 2, pp. 919–926, 2020.

[18] S. Mishra and A. Paul, “A critical analysis of attack detection schemes in IoT and open challenges,” in *Proceedings of the 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 57–62, 2020.

[19] S. Raza, S. Duquennoy, T. Voigt, and U. Roedig, “Demo abstract: securing communication in 6LoWPAN with compressed IPsec,” in *Proceedings of the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011.

[20] S. Zulkarnain and S. Idrus, “Soft Biometrics for Keystroke Dynamics,” in *Proceedings of the International Conference Image Analysis and Recognition*, 2015.

[21] A. H. M. Alaidi, R. A. M. Al_airaji, H. T. ALRikabi, I. A. Aljazaery, and S. H. Abbood, “Dark Web Illegal Activities Crawling and Classifying Using Data Mining Techniques,” *International Journal of Interactive Mobile Technologies*, vol. 16, no. 10, 2022.

[22] G. Zyskind and A. S. Pentland, “Decentralizing Privacy: Using Blockchain to Protect Personal Data,” 2015.

[23] B. Benshoof, A. Rosen, A. G. Bourgeois, and R. W. Harrison, “Distributed decentralized domain name service Proc,” *2016 IEEE 30th Int. Parallel Distrib. Process. Symp*, vol. 2016, pp. 12791287–12791287, 2016.

[24] A. Ouaddah, A. A. Elkalam, and A. Ouahman, “FairAccess: a new blockchain-based access control framework for the Internet of Things,” *Secur. Commun. Networks*, vol. 9, no. 18, pp. 59435964–59435964, 2016.

[25] M. Ali, “Blockstack: A global naming and storage system secured by blockchains,” *USENIX Annu. Tech. Conf*, pp. 181194–181194, 2016.

[26] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things.” IEEE Access.