International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I11P232

Volume 70 Issue 11, 295-304, November 2022

Original Article

An Experiment of a Parallel Entry Security Testing Approach using Ethereum Blockchain

Mansi Bosamia[1], Dharmendra Patel[2]

[1,2] *Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Charotar University of Science and Technology(CHARUSAT), Gujarat, India.*

[1]*Corresponding Author : mansibosamia@gmail.com*

Received: 06 August 2022 Revised: 12 November 2022 Accepted: 16 November 2022 Published: 26 November 2022

Abstract - *The blockchain has had a greater interest in research and the Technical Industry regarding security testing in recent years. A parallel entry security testing approach uses time and security for security enhancement testing. The first entry tests for a normal wallet with basic security, while the other one uses an Ethereum wallet with a Merkel hash tree and Smart Contract. This approach is tested for 100 users and more than 20,000 transactions which shows that Ethereum transactions are faster and more secure using Merkle Hash Tree, while Normal transactions are slower and less secure. Only secure transactions are recorded in the database others are rejected. Thus, the digital wallet security is enhanced using Merkel Hash Tree with an average running time of 1.39.*

Keywords - *Bitcoin, Blockchain, Ethereum, Merkle Hash Tree, Smart Contract.*

1. Introduction

The security testing work is very difficult because identifying faultless and deficient behaviours in computer networks is very tough, as the network limits cannot be well-defined. The blockchain is utilized to diminish this issue. Many blockchain implementations are available in the market, but the most popular is Bitcoin [1]. Bitcoin-based blockchain has the disadvantages such as double-spending, consensus mechanism, and central authority problems. This study used Ethereum with Merkle Tree and Smart Contract to overcome these disadvantages. So, Bitcoin [2] is not only the solution for digital wallets [3].

Blockchain is a structure for storing data in a digital ledger with groups of valid transactions, known as blocks. It makes a consecutive chain with every block cryptographically linked with the preceding block [4-14]. Ethereum is an open-source, blockchain-centred operating system that can be utilized to create smart contract-centric apps that can run decentralized applications based on blockchain technology which uses Merkle Hash Tree and Smart Contract [4, 30]. It has solved double-spending concerns without a central entity. To record the Ethereum blockchain transactions, blocks are used with a static structure. All the transactions are encrypted and kept in the blocks. Merkle hash tree uses the hash function to link the blocks and to test their security. Merkle tree [15] is a tree that has nodes labeled by the hash of its child. To maintain system integrity, Merkle root also indicates the change if any node or single transaction data is altered. Consensus [16-18] is the process of checking that a correct node is joining in transactions blockchain, and the strategy makes the basic validation procedure in the Ethereum blockchain, which prevents central authority problems [19]. Ethereum was designed to be adaptable and flexible, which are properties missing in Bitcoin. Ethereum digital currency unit token known as ETH (Ether) is just a click start for new application developers, investors, miners, and stakeholders. Ethereum works as Bitcoin but can also run and deploy smart contracts [20-25, 30].

Smart contracts are simply programs saved on a blockchain that run when predetermined criteria are satisfied. The code in the smart contract is executed when transactions have happened, and it verifies the complexity depending on the deployed code. The developer of a digital wallet instructs the code. Smart Contracts are used to enforce the rules of transaction interactions that must be followed automatically [26, 27, 30]. Solidity is a statically-typed curly-braces programming language structured for creating smart contracts that run on Ethereum.

In this experiment, the implementation is done for the Ethereum Blockchain instead of the Bitcoin Blockchain and tested for digital security with the proposed MWallet Model. The proposed model does a similar entry for the Ethereum blockchain and Normal Basic Wallet. The comparison analysis between the Ethereum blockchain wallet with Merkle Hash Tree and smart contracts and the Normal Wallet with

basic security. The result shows that the Ethereum blockchain is more secure for Wallet applications than basic web wallet security.

2. Comparison of Bitcoin and Ethereum Merkle Tree Implementations

Merkle hash tree has many functionalities that can be used in its implementations in the blockchain. Table 1 shows the comparison of bitcoin and Ethereum for Merkle tree implementation, which was used to identify the better implementation for the proposed model. Thus, this comparative analysis is given below:

Table 1. Comparison of Merkle tree implantations

Merkle Implementations	Bitcoin	Ethereum
Type	Wallet	Wallet
Blockchain Uses	YES	YES
Open-source	YES	YES
Consistency Verification, Data Verification, Data Synchronization	YES	YES
Decentralized	YES	YES
Distributed Network	YES	YES
Peer-to-peer	YES	YES
Global	YES	YES
Fast	YES	YES
Reliability	YES	Partially
Correctness	YES	YES
Secure	YES	YES
Sophisticated and flexible	YES	YES
Pseudoanonymity	YES	YES
Anonymous	Highly	YES
Automated	YES	YES
Scalable	YES	YES
Platform for integration	YES	YES
Algorithm to generate Hash Value	SHA2	KECCAK-256
Data Mining	Not easy, very difficult	Based on the mining contract
Trusted	YES	YES
De-individualized information	YES	YES

Based on comparison analysis, both implementations satisfy the basic functionality except the reliability, generating hash value and data mining. Due to this comparison results and the Smart Contract [30], Ethereum has more benefits. Thus it is the better solution to use instead of bitcoin for the proposed model.

3. Proposed Model

The proposed model, named MWallet which, do the parallel entry of transactions for Ethereum blockchain-based Wallet and Basic Web Wallet. This comparative entry analysis has been done to prove that the Ethereum blockchain is more secure and faster. Successful transactions involve transferring Payment from one user account to another.

3.1. Major Steps of the Proposed Model

In this architecture, Users must be registered with MWallet. Suppose User A sends a payment request to User B using the User Interface of MWallet. User Interface forwards the Payment Transaction execution time calculation, and Third, Storing. Figure 1 shows Payment Transaction execution, then parallel entry in both Ethereum wallets and Basic Web wallets, calculating the transaction execution time and storing it in the database. Ethereum wallet has Merkle tree-based hashing for security using smart contracts, while the primary web wallet transaction has basic web security. Ethereum blockchain transactions are tested using the Merkle hash tree, while Basic transactions are tested using the Secure Socket Layer [28, 29].

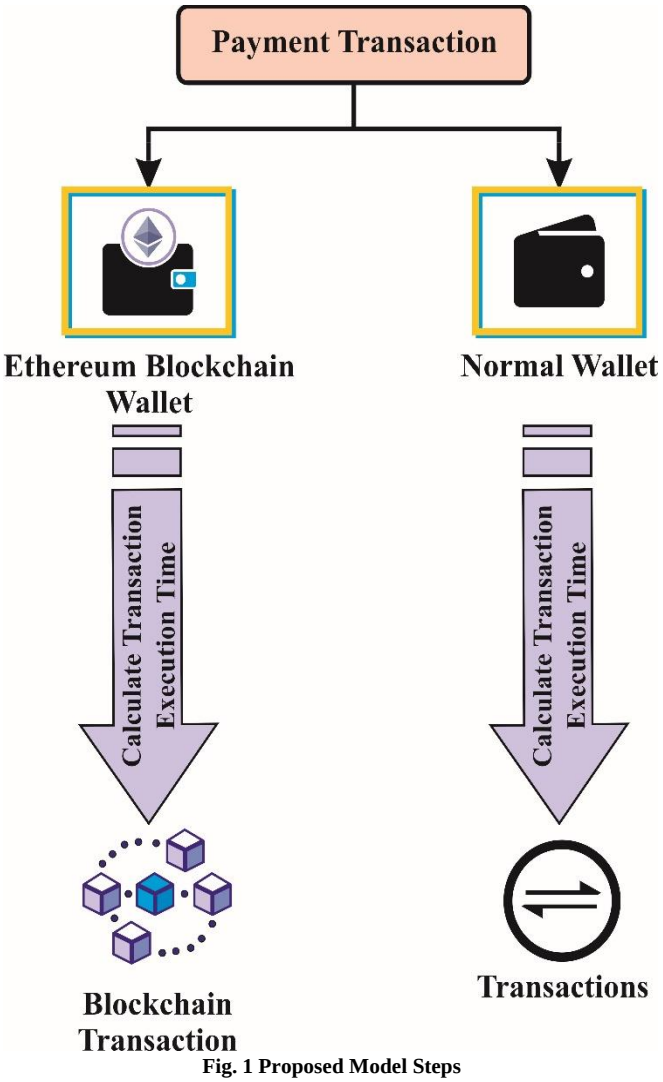

Fig. 1 Proposed Model Steps

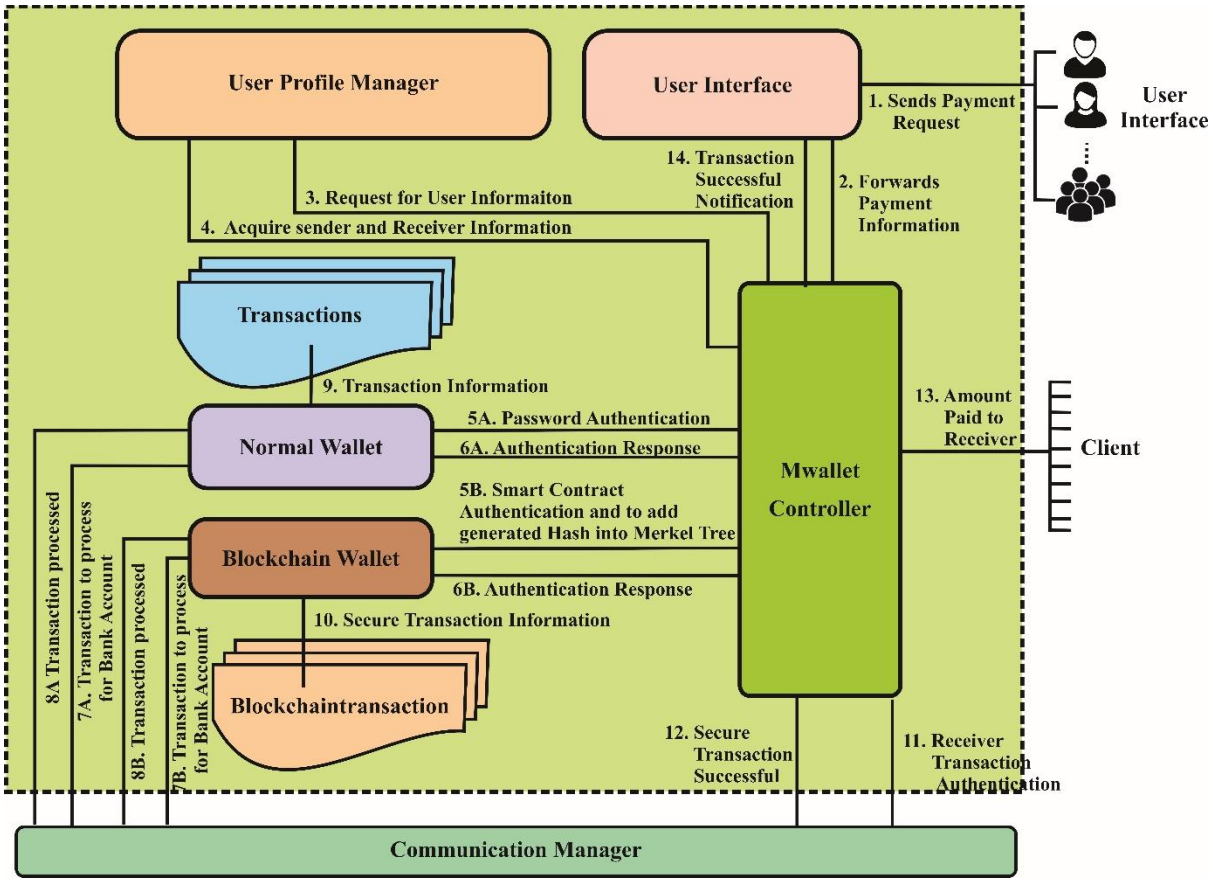

Fig. 2 Architecture of the proposed Model

3.2. The Architecture of the Proposed Approach

Figure 2 shows the proposed MWallet's architecture. In this architecture, Users must be registered with MWallet. Suppose User A sends a payment request to User B using the User Interface of MWallet. The user interfaces the Payment Transaction Information forwards to the MWallet Controller. Figure 2 shows how the MWallet Controller uses the User Profile Manager to authenticate the users (sender and receiver) information and makes similar entries to both the Blockchain wallet and the Normal cum Basic wallet with a calculation of execution time. The standard wallet uses SSL for basic password authentication, whereas the Blockchain wallet uses a Smart Contract with Merkle Hash Tree to improve security. The speed and convenience of working with the wallet are high in the Ethereum Blockchain wallet than in the normal wallet. Smart contracts on Ethereum use Merkle tree-based hashing for security, whereas conventional transactions use basic web security. If both entry is secure, the transaction was successful and made the entry in the respective databases. The Communication Manager is responsible for verifying transaction success. MWallet Control Manager notifies the sender of the transaction's success or failure after verification. For the transaction's success, the sender paid the amount deducted from his/her wallet account and added it to the receiver's wallet account only once. Notify both the sender and the receiver of the same. The MWallet Controller provides an interface to all the services that the MWallet may offer to external objects. The "outside world" cannot see and does not have direct access to any of the components inside the wallet.

In the proposed model implementation, the Blockchaincontroller.js file contains all the methods to perform the MWallet controller jobs. Once the MWallet Controller method is invoked, the MWallet Controller coordinates the steps that need to be carried out among the User Profile Manager (Usercontroller.js) and Smart Contract to execute to make a payment or other operation. The MWallet Controller makes the appropriate calls to the User Profile Manager to ensure that the user involved has the appropriate privileges to carry out the operation.

The User Profile Manager stores information about registered users of MWallet. The proposed model is done the comparative analysis, so the smart contract is used to authenticate one or more financial instruments. It is important to note that the existing wallet did not allow the software agents; it required human users and actual banks. However, this section authorized to conduct of financial transactions without an actual bank and MWallet Money transferred to check the security. The financial transactions happened between registered users only to avoid the actual financial

transaction due to achieving the research purpose to check security and time, not money transfer. Although the User Profile Manager provides access to information about which users are authorized to use MWallet money and who "owns" or may access their MWallet account. The MWallet Controller provides synchronized, concurrent access to users' transactions to prevent conflicting operations. An MWallet should also ensure that when two copies are created, the digital cash is spent only once, not doubly, because the same client can make concurrent digital cash payments [3].

3.3. Merkle Hash Tree Algorithm

The algorithmic procedure of the Merkle hash tree in the authentication process is given as follows:

Step 1: Find the position of the data from another server in the list. Probably by searching by id.

Step 2: Estimate the hash of data from another server.

Step 3: Assess the parent node's value by hashing the current node with its neighbor (if the position is odd, it goes to the next node, and if the position is even, it goes to the previous node) and set the parent as the current node.

Step 4: Step 3 is repeated until the root is found.

Step 5: Contrast the new root with the existing root. If the new root matches, then the data from another server is essentially required data and not tampered with.

3.4. The Proposed Model Implementation in Brief

The proposed model implementation used the listed technologies:

- **Node.js**
  - API, HTML, DB interaction
- **Blockchain with Merkle Hash Tree and Smart Contracts**
  - Ethereum: ganache-cli, geth
  - Development Framework: truffle
- **Software Versions**
  - MS SQL 5.7
  - XAMPP Control Panel v3.2.2
  - Ganache 1.2.1
  - Visual studio code 1.26

4. Main Screen Layout of Proposed Model

The proposed model is implemented as MWallet Web Application, and the proposed research is conducted on it. Its screen layouts are given below with brief descriptions.

4.1. Dashboard

This screen is used to show the transaction capacity and both the wallet balance in Figure 3.

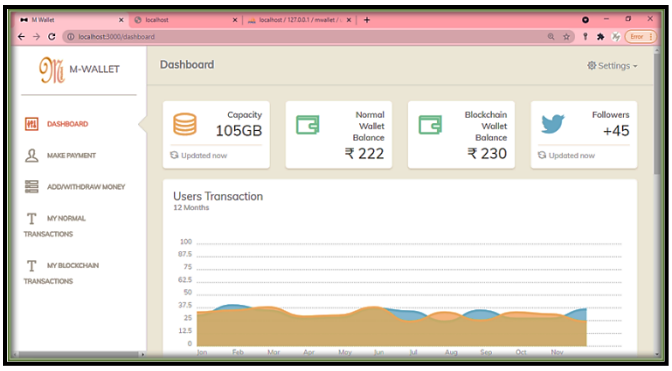

Fig. 3 Dashboard Screen layout

Figure 4 shows the user's successful transactions comparison of both wallets so the researcher can observe the immediate success rate of transactions. This information shows year-wise.

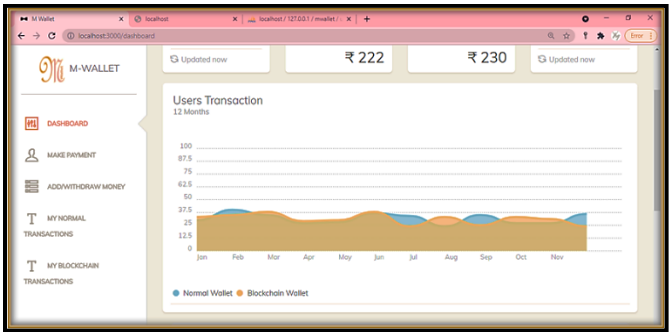

Fig. 4 Dashboard Screen layout for Users Transaction

In Figure 5, the dashboard, this part of the screen shows the average time of secure transactions completed. It also shows the number of Comparison tables of successful transactions. It shows the month-wise graphical analysis of the current year.

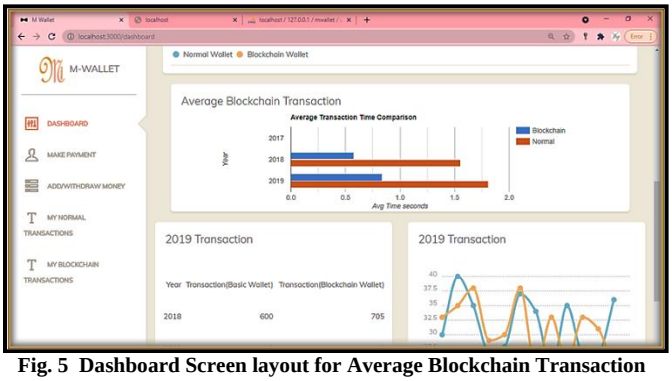

Fig. 5 Dashboard Screen layout for Average Blockchain Transaction

4.2. Normal Transactions

Figure 6 shows the option of MWallet, which is used to show Money In (Received Money) and Money Out (Sent Money) records and All (IN/OUT) both together of Normal Transactions.

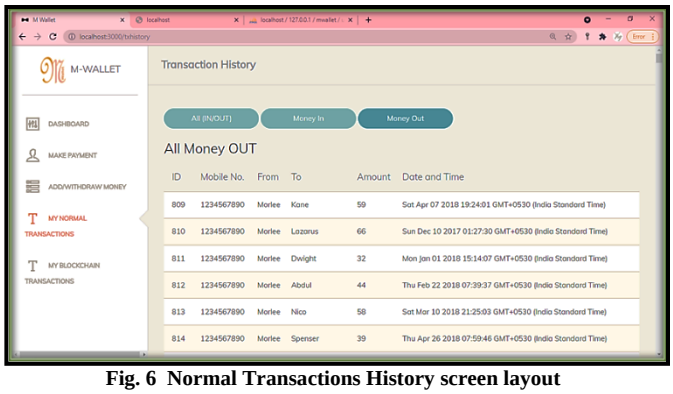

Fig. 6 Normal Transactions History screen layout

4.3. Blockchain Transactions

Figure 7 shows the option of MWallet, which is used to show Money In (Received Money) and Money Out (Sent Money) records and All (IN/OUT), both together of Ethereum Blockchain Transactions with the hash.

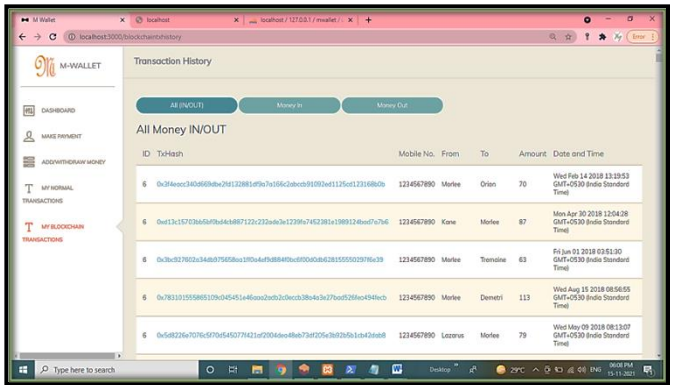

Fig. 7 Blockchain Transactions History screen layout

4.4. MWallet Database

Below, figure 8 shows the blockchain transactions table. To store blockchain Ethereum transactions with the time difference.

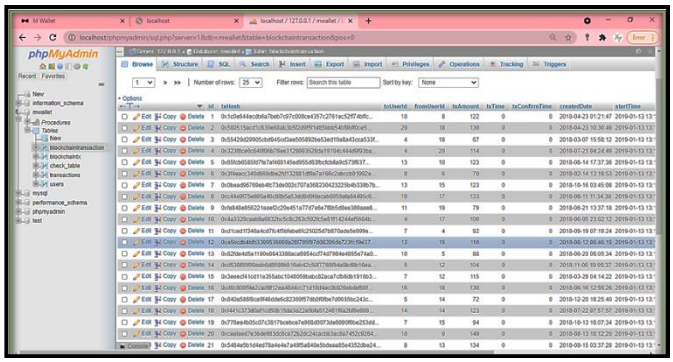

Fig. 8 Wallet Database Screen Layout of the blockchain transaction table

5. Experiments and Results

5.1. Dataset Description

Currently, there are four tables in MWallet Testing Model:

5.1.1. Users

To store transaction user information.

5.1.2. Blockchain Transaction

To store blockchain Ethereum transactions with the time difference.

5.1.3. Transactions

To store Normal wallet transactions with the time difference.

5.1.4. Blockchain tx

To temporarily calculate the transaction difference and update it in the blockchain transaction table.

Based on these data tables, the proposed research ensures the processing, storing, and parallel execution of Basic and Ethereum transactions.

5.2. A Parallel Entry Security Testing Algorithm

This Algorithm is used to test digital wallet transactions with security testing and making parallel entries of secure transactions with Payment only once.

Step 1: Input payment request and response.

Step 2: Retrieve transaction hash.

Step 3: Store the transaction request time.

Step 4: Create two copies of the transaction.

The first transaction copy authenticates using Merkle Hash Tree and Smart Contract.

The second transaction copy authenticates using basic web security.

Step 5: The transaction requires the sender and receiver's mobile number and account details for each Payment.

If the mobile number user successfully authenticates,

If the authenticated mobile wallet has sufficient balance,

Move to step 6.

Otherwise,

The request response is that the transaction is rejected.

Otherwise,

The request response is that the transaction is rejected.

Step 6: Send a transfer authorization from the MWallet to the payment account issuer requesting a money transfer of the transfer amount from an account of the sender's mobile number to the account of the receiver's mobile number only once if it is not rejected.

Step 7: Calculate the Transaction time based on the noted request time and security for Normal wallet and Blockchain wallet transactions.

Step 8: Confirm the money transfer to the payment sender and receiver.

6. Results and Discussion

The user's transactions were recorded from January 2018 through January 2020 to provide test data in Table 2.

1) Experiment results are found from implementing the Ethereum Blockchain web wallet with Merkle hash tree, smart contract, and Normal web wallet with SSL. The web wallet uses the proposed Parallel Entry Security Testing mechanism. It is stated that Merkel hash tree and smart contract 5.79% enhances current digital wallet security.

The user's transactions were recorded over a period from January 2018 through January 2020 to provide test data in Table 2 and its analysis in Figure 9. The average success rate was 94.20%, the reject rate was 5.80% for Normal Transactions, the average success rate was 99.99%, and the reject rate was 0.01% for Ethereum Blockchain Transactions. Table 2 shows the Total Number of Transactions per Month for normal and Ethereum transactions. Ethereum blockchain shows a higher number of transactions for all the months and years, with the highest of 2314 transactions for the month and year of Dec-18. Here, the user's transactions period from April 2019 through January 2020 tested data for more unsecure transactions due to tested with attacks; in this case, this Merkle hash tree-based security allows only secure transactions and others are rejected. So, no entry was found in the database. So, security was tested by changing the Smart Contract.

Table 2. Total Number of Transactions per Month

	Normal Transactions	Ethereum Transactions
Jan-18	507	541
Feb-18	476	460
Mar-18	493	523
Apr-18	479	481
May-18	480	477
Jun-18	498	486
Jul-18	487	507
Aug-18	2255	2293
Sep-18	2244	2264
Oct-18	2226	2267
Nov-18	1722	2166
Dec-18	1823	2314
Jan-19	1864	1936
Feb-19	1625	1626
Mar-19	1732	1732
Nov-19	24	25
Jan-20	1	1
Total	**18936**	**20099**

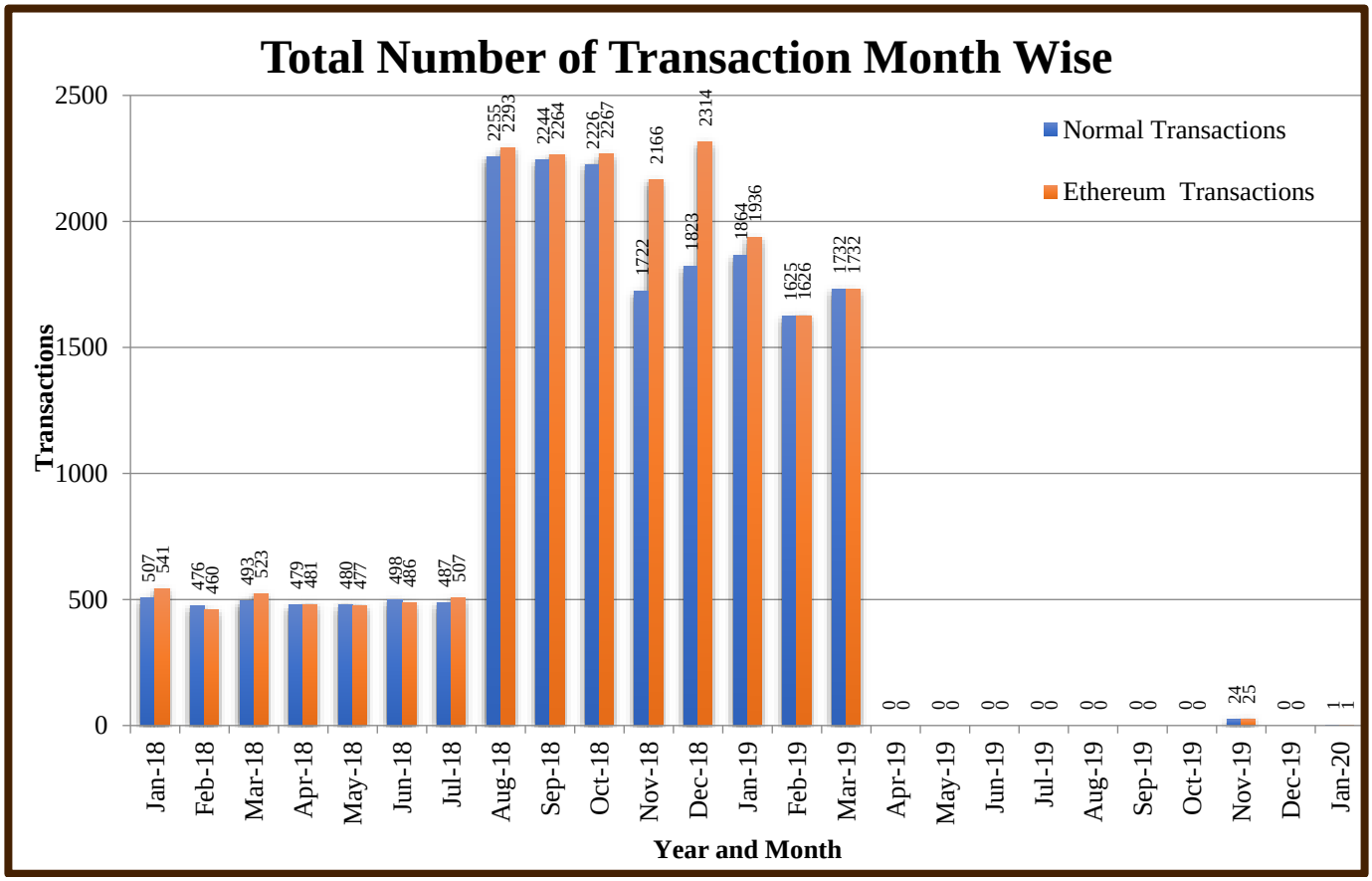

Fig. 9 Month and Year wise transactions analysis

2) Implementation results also state that the Merkel hash tree and smart contract are 0.14 seconds faster than a current digital wallet.

In this situation, the user's transactions from April 2019 to January 2020 were examined for more unsecure transactions owing to attacks. In this case, Merkle hash tree-based security enables only secure transactions and rejects the rest. As a result, there was no entry in the database. The user's transaction time was recorded from January 2018 to January 2020 to provide test data in Table 3 and analysis in Figure 10. The average time for a typical transaction was 1.53, whereas the average time for an Ethereum Blockchain transaction was 1.39. A few months have had no secure transactions due to the alteration of intelligent contracts to test more security.

3) The result of the implementation also states that the Merkel hash tree and smart contract rejects unsecure transaction against clever contract alteration and attacks.

Here, the user's transactions were recorded over a period from January 2018 through January 2020. In 2019, the user's transactions experimented with smart contract alteration to overcome the Merkle hash tree security, but it is more secure than other mechanisms, so it rejects the transaction instead of execution. Thus, this experimental result states that the Merkle hash tree gives more security when altering the smart contract. While in the current digital wallet, security generates an error, and due to exception handling, data entry is not recorded. Thus, April 2019 through January 2020 tested data entry was not done in the database. The average success rate was 94.20%, the reject rate was 5.80% for Normal Transactions, the average success rate was 99.99%, and the reject rate was 0.01% for Ethereum Blockchain Transactions**.** Here, success rate means secure transactions are done while others are rejected. In 2020, only 1 transaction because this year, the researcher changed the smart contract for more security to achieve. Due to that, only 1 transaction they are securely done.

Table 3. Transaction Average Time to Perform the Transactions

Month and Year	Normal Transactions	Ethereum Transactions
Jan-18	1.55	1.32
Feb-18	1.43	1.31
Mar-18	1.50	1.38
Apr-18	1.53	1.38
May-18	1.34	1.51
Jun-18	1.39	1.30
Jul-18	1.54	1.41
Aug-18	1.66	1.67
Sep-18	1.95	1.64
Oct-18	1.55	1.85
Nov-18	1.71	1.63
Dec-18	1.69	1.61
Jan-19	1.56	1.74
Feb-19	1.33	1.67
Mar-19	1.77	1.72
Nov-19	1.52	0.44
Jan-20	1	0
Average Transaction Time	**1.53**	**1.39**

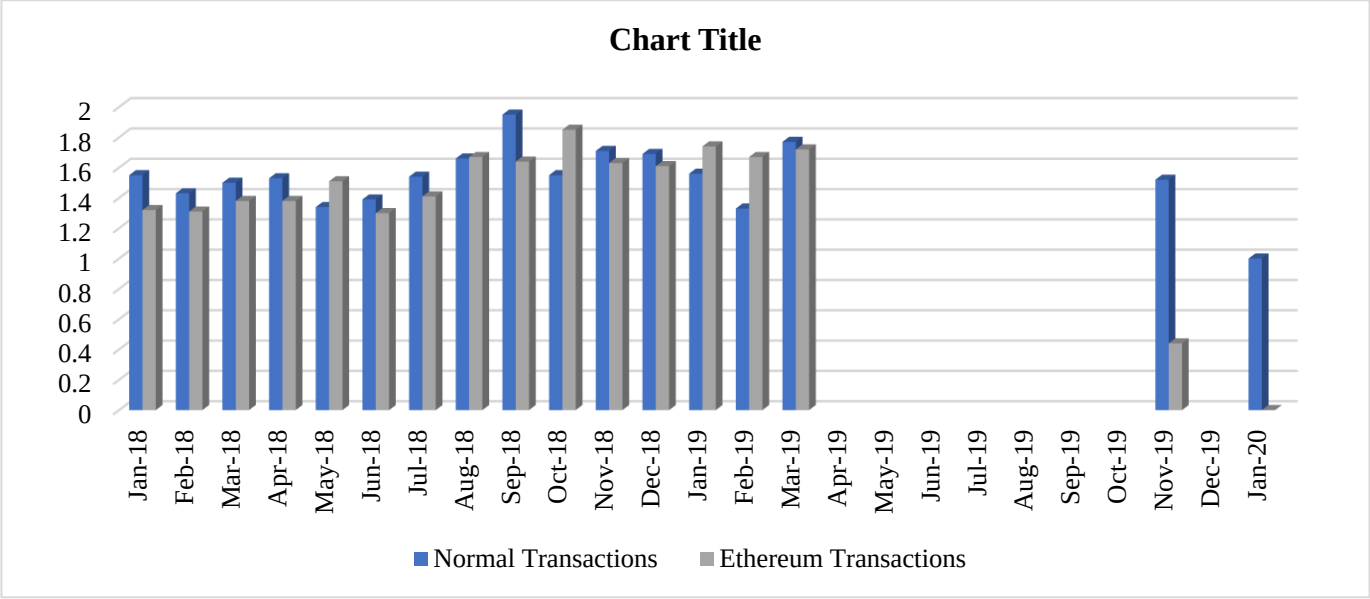

Fig. 10 Transaction Average Time to Perform the Transactions

Table 4. Total Number of Transactions per Year

	Normal Transactions	Ethereum Transactions
2018	13690	14779
2019	5245	5319
2020	1	1
Total	**18936**	**20099**

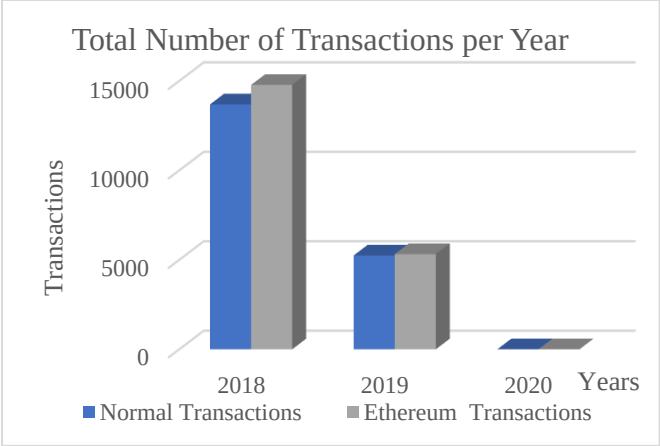

Fig. 11 Year-wise total numbers of transaction

7. Comparisons of Ethereum Blockchain and Normal Digital Wallet Transactions

As mentioned in Table 5, the proposed system works for the Ethereum blockchain with Merkle Hash Tree, and Smart Contract has a secure success rate of 99.99% on average 1.39 processing time while Normal with SSL has a secure success rate was 94.20% in an average 1.53 processing time. It shows that Merkle Hash Tree with Smart Contract gives secure and faster results than SSL.

Table 5. Transactions with Merkle Tree experiment sample set-up details

Experiment Sample Set	Number of Transactions	Security Mechanism	Secure Transactions Rate	Unsecure Transactions Rate	Average Transaction Time
Normal digital wallet transactions with Basic Security	20102	Secure Sockets Layer (SSL) protocol [27,28]	94.20%	5.80%	1.53
Ethereum Blockchain digital wallet transactions with Merkle Hash Tree and Smart Contract Security	20102	Merkle Hash Tree, Smart Contracts	99.99%	0.01%	1.39

8. Contribution

The new parallel entry approach has been developed and tested for Ethereum Blockchain with Merkle Hash tree with Smart Contract and Basic Web Wallet with basic web security. A comparative analysis has been done, which shows that Ethereum is faster and more secure than basic SSL security. A parallel entry approach can be used for other comparative analyses like the Ethereum blockchain vs Bitcoin blockchain. The proposed model is used for any similar entry of security measures with the Ethereum blockchain, and the researcher can achieve successful results.

9. Conclusion

A parallel entry security testing experiment has been discussed in this paper. The web wallet transactions are processed by it. The efficient result in a comparison of Ethereum blockchain transactions using Merkle Hash Tree with Smart Contract security and Basic transactions with basic security shows that Ethereum blockchain with Merkle Hash Tree & Smart Contract is 0.14 seconds faster and 5.79% more secure. The experiment is conducted on 20102 transactions by various 105 users. The system can be further exploited by testing more samples with different attacks. Also, smart contract alteration can enhance the average secure and faster transaction rate, depending on the application implementation.

Future Scope

A parallel entry security testing algorithm can be used and tested for other market solutions like medical records, banking transactions, notary services, user's identity verification and status, data tracking, etc., concerning security, privacy, efficiency, transparency, and fault-resistances.

Acknowledgments

We thank Dr. Atul Patel and Dr. Darshan Choksi for many thoughtful comments on my research work. The authors sincerely thank Charotar University of Science and Technology (CHARUSAT) for giving them the required resources to undertake this experiment.

References

[1] J. P. M. Cruz, "The Bitcoin Network as Platform for Role-Based Access Control And Electronic Voting Using Blockchain-Based Technology to Create Innovative Systems," *Doctoral Dissertation, Nara Institute of Science and Technology*, 2017. Crossref, https://doi.org/10.34413/dr.01404

[2] Decker C, "On the Scalability and Security of Bitcoin," *Doctoral Dissertation*, ETH Zurich, 2016. Crossref, https://doi.org/10.3929/ethz-a-010619000

[3] Daswani N, Boneh D, Garcia-Molina H, Ketchpel S. P and Paepcke A, "SWAPEROO: A Simple Wallet Architecture for Payments, Exchanges, Refunds, and Other Operations," In *USENIX Workshop on Electronic Commerce*, 1998.

[4] Bosamia M and Patel D, "Comparisons of Blockchain-Based Consensus Algorithms for Security Aspects," *International Journal on Emerging Technologies*, vol. 11, no. 3, pp. 427-434, 2020.

[5] Wood G, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1-32, 2014.

[6] Hamida E. B, Brousmiche K. L, Levard H and Thea E, "Blockchain for Enterprise: Overview, Opportunities and Challenges," In the *Thirteenth International Conference on Wireless and Mobile Communications (ICWMC 2017)*, 2017. https://hal.archives-ouvertes.fr/hal-01591859

[7] Bashir I, "Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained," 2nd Edition, ISBN: 978-1788839044, 2018.

[8] Zheng Z, Xie S, Dai H. N, Chen X, and Wang H, "Blockchain Challenges and Opportunities: A Survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352-375, 2018. Crossref, https://doi.org/10.1504/IJWGS.2018.10016848

[9] Kulkarni S, "The Beauty of the Blockchain," *Open Source for you, the Complete Magazine on Open Source*, vol. 6, no. 8, pp. 22-24, 2018.

[10] Mattila J, "The Blockchain Phenomenon," *Berkeley Roundtable of the International Economy, Working Paper*, 2016. [Online]. Available: http://brie.berkeley.edu/BRIE/

[11] Yaga D, Mell P, Roby N, and Scarfone K, "Blockchain Technology Overview," 2019. arXiv preprint arXiv:1906.11078.

[12] Aixa D. R, "Analysis and Study of Data Security in the Internet of Things Paradigm from a Blockchain Technology Approach," *Blockchain Technologies and Application*, 2018. [Online]. Available: http://hdl.handle.net/10609/72949.

[13] Jiang H, Liu D, Ren Z, & Zhang T, "Blockchain in the Eyes of Developers," *ArXiv preprint arXiv:1806.07080*, 2018. https://doi.org/10.6028/NIST.IR.8202

[14] Satish Chandra Gullena, "Iot Architectures Based on Blockchain Technologies," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 7, pp. 874-878, 2018.

[15] Baliga A, "Understanding Blockchain Consensus Models," Persistent 4, pp. 1-14, 2017. [Online]. Available: https://www.linkedin.com/pulse/understanding-blockchain-consensus-models-arati-baliga

[16] Zheng Z, Xie S, Dai H, Chen X, Wang H, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," In *2017 IEEE international congress on big data, BigData Congress, IEEE,* 2017. Crossref, https://doi.org/10.1109/BigDataCongress.2017.85

[17] Sankar L. S, Sindhu M, and Sethumadhavan M, "Survey of Consensus Protocols on Blockchain Applications," In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE,* pp. 1-5, 2017. Crossref, https://doi.org/10.1109/ICACCS.2017.8014672

[18] Shah T, and Jani S, "Applications of Blockchain Technology in Banking & Finance," *Parul University*, 2018. https://doi.org/10.13140/RG.2.2.35237.96489

[19] Sajana P, Sindhu M and Sethumadhavan M, "On Blockchain Applications Hyperledger Fabric and Ethereum," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 2965-2970, 2018.

[20] Chan W and Olmsted A, "Ethereum Transaction Graph Analysis," *12th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 498-500, 2017. Crossref, https://doi.org/10.23919/ICITST.2017.8356459

[21] Gencer A. E, Basu S, Eyal I, Renesse R. V, and Sirer E. G, "Decentralization in Bitcoin and Ethereum Networks," In *International Conference on Financial Cryptography and Data Security, Springer,* pp. 439-457, 2018.

[22] Vujičić D, Jagodić D and Ranđić S, "Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview," *In 2018 17th International Symposium Infoteh-Jahorina (Infoteh), IEEE,* pp. 1-6, 2018. Crossref, https://doi.org/10.1109/INFOTEH.2018.8345547

[23] Dannen C, "*Introducing Ethereum and Solidity*," Berkeley Apress, 1st Edition, ISBN: 978-1-4842-2535-6, 2017.

[24] Blockchain and Payment. [Online]. Available: https://in.linkedin.com/in/ramalingom-sundaram-pillai-343a13120

[25] Understanding the DAO Attack. [Online]. Available: http://www.coindesk.com/understanding-dao-hack-journalists/

[26] Christidis K and Devetsikiotis M, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292-2303, 2016. https://doi.org/10.1109/ACCESS.2016.2566339

[27] Buterin V, "A Next-Generation Smart Contract and Decentralized Application Platform," *Ethereum White Paper*, vol. 3, no. 37, pp. 1-2, 2014.

[28] Young A, Chapman P and Berardy R, U.S. Patent Application No. 09/728,471, "System and Method for Performing an Electronic Transaction Using a Transaction Proxy with an Electronic Wallet," 2002. [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2001041419

[29] Bhite N and Sharma M. A, "The Algorithm Analysis of Electronic Payment Systems," *International Journal of Scientific Engineering and Research*, vol. 4, no. 9, pp. 4-6, 2016.

[30] R. Sujeetha and C. A. S. Deiva Preetha, "Analysis on Mutation Testing Tools for Smart Contracts," *International Journal of Engineering Trends and Technology*, vol. 70, no. 9, pp. 280-289, 2022. Crossref, https://doi.org/10.14445/22315381/IJETT-V70I9P228

[31] P. Thirugnanam, M. Pavithra and B.Akoramurthy, "Decentralized Server Using Bitcoin Cryptography and Bittorret Network," *International Journal of Engineering Trends and Technology*, vol. 45, no. 1, pp. 10-13, 2017. Crossref, https://doi.org/10.14445/22315381/IJETT-V45P203