IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Current Trends and Future Implementation Possibilities of the Merkel Tree

Mansi Bosamia[1*], Dharmendra Patel[2]

[1]Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, CHARUSAT, Changa, Gujarat, India
[2]Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, CHARUSAT, Changa, Gujarat, India

Corresponding Author: mansibosamia@gmail.com, Tel.: +91-94280-79200

Available online at: www.ijcseonline.org

Accepted: 16/Aug/2018, Published: 31/Aug/2018

***Abstract*—** A current popular trend of wallet security research is cryptography. For that Merkle tree is one of the solutions to enhance wallet security. It is basically used for cryptocurrencies, file system authentication, backup system, control system, database, etc. but it can be used for communication authentication and many more. And that is highlighted in this paper by stating the few future Merkle tree implementation possibility with its basic technical requirements. In this survey study also discuss about the Merkle tree concept with its advantages and disadvantages and its implementations such as Bitcoin, Ethereum, Hash-based Cryptography, Apache Cassandra, Btrfs, ZES, IPFS with their comparisons.

***Keywords*—** Merkle Tree, Bitcoin, Ethereum, Hash-based Cryptography, Apache Cassandra, Btrfs, ZES, IPFS

I. INTRODUCTION

Considering the fact that now a day everyone needs their data secure on devices, application, network and databases. So the current trends of research are wallet security enhancement. To enhance the security in current time one of the solutions is Merkle tree. Merkle tree is data structure used for cryptocurrency with hashing algorithm. Its popular implementation is Bitcoin. This article study contains the basic concepts of Merkle tree, Merkle tree node, its advantages and disadvantages with its current popular implementation's basic information and functionality based comparisons. Also identify the basic technical parameters for future Merkle tree implementation and suggest the combination of GUI, Sensor detection and NFC based innovative system of merkle tree implementations in wireless environment.

This paper organized as follows, Section II contains the Merkle tree concept which explain using an example and its advantages and disadvantages, Section III contains the Merkle tree implementations like Bitcoin, Ethereum, Hash-based Cryptography, Apache Cassandra, File Systems, Section IV contains the comparison of merkle implementations based on its functionality for identifying the future applications, section V explain future Merkle tree implementation possibilities, Section VI describes technical requirements for Merkle tree implementations, and Section VII contains conclusion with future directions of this paper.

II. MERKEL TREE

In computer science applications and cryptography, a tree *data structure used for cryptocurrencies is a "Merkle tree"*. It is also called *"Binary Hash tree"*. Binary hash tree has given named Merkle tree by Ralph Merkle in 1987 paper titled "A Digital Signature Based on a Conventional Encryption Function." This concept is patented by Ralph Merkle in 1979 and proposed the cryptographic hashing. [17, 18] Merkle tree has hashing paired data as tree leaves or node, then pairing and hashing the results until the single hash remains as the Merkle root. Merkle root is the root node of the Merkle tree with a successor of all the nodes in the tree. In a Merkle tree context, record may refer the word "transaction". In a Merkle tree transaction is a packet of data whose hash communicates to a leaf node. All the permanent transaction data records in files that represents as the Merkle tree leaves is called blocks. The constant transactions are the tree leaves from a single block access by Merkle root. In Merkle tree each leaf node has the block data hash and each non-leaf node has its child nodes cryptographic hash. Merkle tree manages the complete account ledger of each user's transactions. See the example of Merkle Tree in Fig. 1

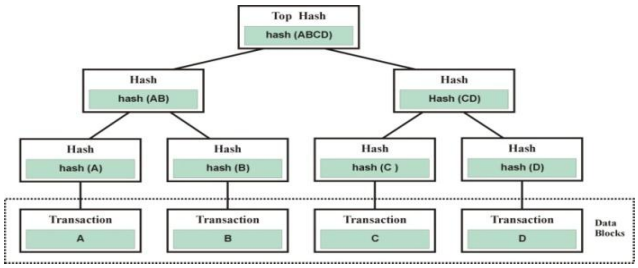

Figure 1. An example of Merkle Tree

The structure of Merkle tree shown in Fig. 1, in this example it contains Transactions ABCD and Hash. This figure is very basic; an average block generally works with 500 or more transactions, not just four. All the bottom nodes are leave nodes and intermediate hash nodes are branches and top hash indicates the Merkle root. This kind of figure through leaf nodes of the Merkle tree calculates the number hashes proportional to the logarithm while the number of proportional leaf nodes has the hash lists [13]. Merkle root address stored in the block header such as #492920. This is not the actual block address its hash of next block. And next block contains another Merkle root which address of its next block, and so on. Merkle tree has number of nodes and each Merkle tree node contains Target hash, Block header, Block time, Hashed time lock contract proofs and next node pointer; refer Fig. 2 [9]. And the Merkle tree root node address is stored in the block of the blockchain.

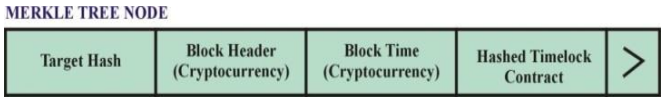

Figure 2. Merkle Tree Node

Here, brief details of Merkle tree node attributes:

[1]. ***Target Hash:*** It is a number that is a hashed block header have to be less than or equal to in order for new block to be granted. It is used to identify the input problems and to adjust the order properly to verify that blocks are processed capably.

[2]. ***Block Header:*** Its size is 80-byte single block to hash continuously to generate the proof of work. It contains the valid Merkle root successor from all the transaction in that block to identify the specific block from complete blockchain.

[3]. ***Block Time:*** The total amount of time to process the block — time start from pushing block from departure (off-block) to received at destination (on-block). The block time may vary according to routes of communication network.

[4]. ***Hashed Time Lock Contract (HTLC):*** It is cryptocurrency channels smart contract to remove the counterparty risk. It allows the time-bound transactions. In practical terms, to allow the transaction, if the payment transaction acknowledgement done by cryptographic proof in definite timeframe otherwise not.

Log is a one kind of Merkle Tree, which build up from the hashed records. It has the property for new entry is consistently added at the last leaf in the tree. One more property is that once records is logged, entry can't changed, if you change then it consider as new record entry in the log. Also gives the entire audit of each transaction. Equally, hash tree database allows adding, editing, and removing the records from whole tree. When a Merkle tree added to the log, Merkle tree contains total number of leaves with the root hash. Thus, a Merkle tree is used to quickly and efficiently identify the changes records to synchronize them in distributes system. [12]

Basically Merkle tree used to authenticate the stored data, to manage data and to transmit between computers. These can verify the received data blocks from senders in peer-to-peer network that data are received properly, damaged, changed and real blocks not fake one. Generally, Merkle tree uses a cryptographic hash function for the hashing. To data verification sometimes it takes support of checksum method. [13]

There are the some advantages and disadvantages of Merkle tree:

Advantages

[1]. Gives the large data structures content secure verification efficiently.

[2]. Maintains consistency, data verification and data synchronization by decreasing the network input/output packet size.

[3]. Allows users to authenticate the particular transaction which exists in the block without accessing the all blocks in the list.

[4]. Increasing data integration and validation.

[5]. Requires small amount of memory.

[6]. Proofs computation is very easy and fast.

[7]. Requires little and brief information on network for proofs and management.

[8]. Provides secure information management by a dictionary mapping tree roots and the leaf counts.

[9]. Supports Simplified Payment Verification (SPV) clients, blockchain pruning and smart pool miners.

[10]. Quickly and efficiently identify the changed records to synchronized them in distributes system.

[11]. By the root hash, easy to identify a particular node. Particular node is m^2 or the child pairs for reconstructing the intermediary root hash.

[12]. Exclusively responsible for appending trees, because the specific system knows the number of leaves in each tree from which adding a node or tree.

[13]. Gives overview of hash chains/lists.

[14]. Merkle tree can be implemented for itself data verification as well as for data store data verification with separate validation of the data.

Disadvantages

[1]. Suffers from second-preimage attack because of it is not able to identify the depth of the tree and attack by accessing same Merkle hash root for transaction.
[2]. Trusted authority maintains the proof of integrity of the data amount is become lower.
[3]. For a consistency proof refers the first *m* leaves in the tree. When we add a tree, *m* leaves are combined for hashes the data verification in order not for changed.
[4]. When the other participants adding a node or a tree in existing Merkle tree, necessary to notify about number of leaves to perform consistency check.
[5]. Merkle signatures are not one-time, thus it uses the private key to sign many messages and this has potentially large finite limit.

III. MERKLE TREE IMPLEMENTATIONS

Most used Merkle tree implementations are digital currency, hash-based cryptography, apache Cassandra, and file system. Merkle tree can also be implemented in global supply chain, capital market, Git and Mercurial distributed revision control system, Zeronet, Dat Protocol, Apache Wave Protocol, the Tahoe-LAFS backup system, the certificate transparency framework, tiger tree hash and many more. Health care industry is upcoming use of Merkle tree. It is used based on Bitcoin for the NHS and the patient's data track in real-time. For example, Deep Mind Health (planning to use), it has digital ledger for verifiable data audit of patient's interaction data in cryptographic manner. That means patients data can be accessible with changes. [12] In this study, most used Merkle tree implementations are as below:

A. Digital Currency

Most of digital currency are cryptocurrency and its plays an important role in economics. In future may bank to be a technically digital savvy bank. To avoid a troublesome and fraud, blockchain technology is one of the solution of it [6]. Blockchain based digital currencies uses the secure decentralized network. As per the measurement study on different decentralization metrics there are two leading digital currencies are Bitcoin and Ethereum [1].

Bitcoin

Bitcoin is most popular unique digital cryptocurrencies in the current e-wallet market. Satoshi Nakamoto has been done the primary Bitcoin implementation of Merkle trees. It uses the hash function's compression step to an excessive degree and that is moderated by using Fast Merkle Trees [13]. According to original Bitcoin paper [10], "Simplified Payment Verification" procedure can be done based on accessing corresponding Merkle branch instead of whole block [7]. Bitcoin is open-source software and its system runs on a distributed decentralized peer-to-peer network. Bitcoin is a subset of digital currency which processes the transaction using the Internet power [6]. In Bitcoin implementation, block header uses Merkle root to calculate all the transaction in blockchain. Merkle tree calculates hashes of transaction and put those pair wise in nodes. It repeats many times to generate a blockchain. Each Bitcoin clients uses Merkle tree. It gives better security in low bandwidth [7].

Bitcoin is public-key cryptography uses the SHA256 hash function to authenticate the Bitcoin transaction by digital signatures. This hash function is used for generating Bitcoin addresses, verifying the payments and signing the transactions. Bitcoin address is an alphanumeric sequence of characters due to cryptocurrency to secure the sender and recipient identify in the Bitcoins. Thus, Bitcoin is an anonymous currency that kind of misunderstood leads. Bitcoin has benefits such as speedy transactions at low costs, freedom of payments (with lack of restriction on transaction, user has liberty to send and receive Bitcoins anyone, anywhere at any time), merchant benefits (electronic payment accepted by business, so customers have not to pay for various fees for transaction and cost effective for merchants). Bitcoin has risks of Bitcoin stolen (to avoid by user control on Bitcoin), internal change and volatility, possibility of criminal activity due to pseudo-anonymity and easiness of payments, economic risk as its novel use may be troublemaking to the financial payment markets. If Bitcoin does the cryptocurrencies mining, then there is possibility of cryptocurrency attack. But without mining using digital register it works very fine. This problem accounted in mining pool so preferably to avoid mining in Bitcoin. [6]

Bitcoin functions on a proof-of-work basis. Proof-of-work means to create blocks and adding in blockchain to solve very difficult mathematical problems. By the proof-of-work Bitcoins security and validity increases with negative effects such as don't have to stake results the malicious activity, large amount of energy require validating transaction. [4]

Ethereum

Ethereum is blockchain platform which is open and to build decentralized peer-to-peer applications that run using blockchain technology. Like Bitcoin, Ethereum is open-source software, easily adaptable and more flexible [5]. Ethereum blockchain is an extremely powerful shared global infrastructure that can exchange the cryptocurrencies value around and represent the ownership of property. Ethereum allows building secure crypto-assets using smart contracts such as censorship, fraud or third party interference, and without any possibility of downtime. It creates a tradable digital token to use as a currency, an asset, a virtual share, a membership proof. These tokens use typical coin API to make any wallet compatible automatically. A total tokens amount can be set to fixed amount or alter the amount based on programmed rule set. Ethereum blockchain is like public ledger of all occurred transactions and each transaction

blocks are added in a linear chronological order and execute and verify it. Due to the public ledger full blockchain cannot be fallacious by single entity. [4] Ethereum used design of Nakamoto consensus and the GHOST protocol for sequencing transactions. Block hash used for responding the client's blocks request and older clients blocks request consist a body and header that cannot request individually. [1]

Ethereum was considered to be much more than a payment system. Ethereum's protocol is built for Ethereum system to increase functionally of various smart contracts in flexible manner. To provide these facility blocks validators will take a transaction fee for validating smart contract on each transaction. Ethereum's protocol focuses on bandwidth rather than hash rate. [4] The objective of Ethereum is to bring together the scripting, altcoins and on-chain meta-protocols to create random consent based applications with scalability, standardization, completeness, ease of interoperability and ease of development. Ethereum does this by Turing-complete programming language with a blockchain at foundational layer to write own smart contract for decentralized applications. Ethereum blockchain blocks contains the copy of the both most recent state and the transaction list. [8] The Ethereum platform through easily the new applications creation possible and with the Homestead release, anyone can use those applications safely [5].

B. Hash-based cryptography

Hash-based cryptography is the universal phrase for creation of cryptographic primitives supported for the security of hash functions. It is restricted to digital signatures scheme as Merkle signature scheme. It is uses combined one time signature and a Merkle tree structure. One-time signature key can sign a single message securely but the merkle signature scheme can sign more than one message securely to expand the structure. In this hierarchical data structure, to compute tree nodes a hash function and concatenation are used frequently. An example of this combined structure is Lamport signatures. Hash-based cryptography is a kind of post-quantum cryptography and the security if other post-quantum cryptographic schemes like lattice-based still needs the further research. XMSS (eXtended Merkle Signature Scheme) is hash-bashed signature and from the 2007, it is a Generalized Merkle Signature Scheme (GMSS). [14] Hash-based signatures were initially proposed by Merkle in the late 1970s. Hash-based signatures have a potential replacement for recent signature schemes when large-scale quantum computers are to be built. There are many reasons for this such as to provide reliable security against the attacks. Hash-based signature differs with other post-quantum signature schemes due to no expensive mathematical operations computation. It is requires just secure cryptographic hash function. [11] Hash based signatures used for state management, authentication, etc.

C. Apache Cassandra

Apache Cassandra is an open-source distributed NoSQL Database Management System (DBMS) which is freely available with easy data distribution. It is created at Facebook and it is different from Relational Database Management Systems (RDMS). Its data model on Google's Bigtable and its distribution design is based on Amazon's Dynamo. Apache Cassandra is a decentralized, column-oriented database, linear scalable, flexible, fault-tolerant, consistent DBMS. It supports ACID (Atomicity, Consistency, Isolation, and Durability) properties. It is designed to handle huge amount of data transversely and numerous commodity servers with high availability without compromising the performance and without single point of failure. It presents stout support for clusters across numerous data centers and permits low latency process for all clients with asynchronous master less duplication. It introduced the Cassandra Query Language (CQL) with easy accessing interface and Structured Query Language (SQL) support. [15] Apache Cassandra is being used by some popular and biggest companies such as Facebook, Twitter, ebay, Cisco, Rackspace, Netflix, and more. Simply, to handle big data workloads, the systems use the NoSQL database and Apache Cassandra supports it. Due to the big data management Apache Cassandra is leading distributed database with zero downtime, linear scalability and seamless multiple data center deployment. It is widely accepted due to large number of online transaction processing by web companies to introduced practical data modeling approach with efficient schema design [7].

D. File System

Merkle tree is also implemented as file system such as Btrfs, ZES, IPFS, etc.

B-tree F S (Btrfs)

Btrfs is a file system based on copy-on-write theory. It is designed at Oracle Corporation for use in Linux file System. It overcomes the limitation of pooling, integral multi-device spanning, snapshots, and checksums. The Linux scale the storage with clean interface reliably by all users. Btrfs is automatic defragmentation and scrubbing features. Due to the copy-on-write nature it is generally self-healing in some configuration. It supports online balancing, online volume growth and shrinking, online block device adding and deletion, offline file system check, file cloning, sub volumes snapshots, transparent compression, atomic writable or read only, block discard, incremental backups, out-of-band data duplications etc.

Zettabyte File System (ZES)

ZES is a file system and logical volume manager designed by Sun Microsystems and now owned by Oracle Corporation to counter data degradation. It is open source data validating enterprise file system. It is scalable with protection against

data corruption, supports high storage capacities, efficient data compression, integration of file system, volume management, snapshots and copy-on-write clones, continuous integrity check and automatic repair, etc. ZES implemented within UNIX like systems.

Inter Planetary File System (IPFS)

Juan Benet designed the IPFS. It generates a content-addressable in peer-to-peer distributed file system for storing and sharing hypermedia. It looks for each and every one computing devices to connect with the same system of files. Currently, used for open-source project by the community. Its infrastructure has store for unalterable data, remove duplicate files from the network and find out the node from the address for searching a file in the network. IPFS has a self-certifying namespace, distributed hash table, and an incentivized block exchange. It has trust on connected nodes only and no single point of failure. It prevents form DDoS attacks and saves the distributed content delivery bandwidth. It has strong limitation of notable users who able block the site as well as contents of particular websites. IPFS is like the Web in some ways but it could be view as single Bit Torrent swarm to swap objects within one Git repository. It gives content-addressed block storage model with a high throughput on content addressed hyper links. This creates a general Merkle DAG which is a data structure based on blockchain, versioned file systems and a permanent web. [9]

IV. COMPARISON OF MERKLE TREE IMPLEMENTATIONS BASED ON ITS FUNCTIONALITY

This comparison is based on merkle implementations functionality, to identify the better implementation to use for better combination with future applications. To analyse them refer the below table 1.

Table 1. Comparison of Merkle tree implantations based on its functionality

Merkle Implementations	**Bitcoin**	**Ethereum**	**Hash-based Cryptography**	**Apache Cassandra**	**Btrfs**	**ZES**	**IPFS**
Type	Wallet	Wallet	Merkle Signature Scheme	NoSQL DBMS	File System	File System	File System
Blockchain Uses	YES	YES	YES	NO, but application combination may use.	Partially, if needed by application	NO	NO, but application combination may use.
Open-source	YES	YES	Implementation may be	YES	YES	YES	YES
Consistency Verification, Data Verification, Data Synchronization	YES	YES	YES	YES	YES	YES	YES
Decentralized	YES	YES	YES	YES	YES	-	YES
Distributed Network	YES	YES	Distributed over signature generations	YES	YES	YES	YES
Peer-to-peer	YES	YES	Based on communication architectures	YES	YES, based on request	-	YES
Global	YES	YES	NO	May be due to Based on application use	Partially, if needed by application	For Database File System	YES
Fast	YES	YES	-	YES	YES	YES	Make the web faster
Reliability	YES	Partially	YES	YES	NO	YES	-
Correctness	YES	YES	Verifies	Ensures	YES	YES	Verifies
Secure	YES	YES	Based on hash function used	YES	YES	-	YES
Sophisticated and flexible	YES	YES	Complex	YES, Also dynamic data model	YES	-	Complex and flexible to use
Pseudo anonymity	YES	YES	YES	YES	YES	-	YES
Anonymous	Highly	YES	YES	YES	YES	-	YES
Automated	YES	YES	Automated analysis possible	Automated test builds	NO	YES	YES

Scalable	YES	YES	NO	YES	YES	YES	YES
Platform for integration	YES	YES	Need to new combined design	Cross-platform, mostly use with Hadoop platform	Easy	Different as per need	Different as per web application need
Algorithm to generate Hash Value	SHA2	KECCAK-256	RSA, DSA and ECDSA	Consistent hashing algorithm, murmur hash algorithm to generate tokens.	SHA-256	Consistent hashing algorithm	different hash algorithms are used
Data Mining	Not easy, very difficult	Based on mining contract	Possible	Possible	Possible	Possible	Possible
Trusted	YES	YES	YES	YES	Partially	YES	Partially
De-individualized information	YES	YES	Based on keys	NO	NO	NO	NO

Based on comparison analysis identify that each implementation satisfy the basic functionality such as consistent verification of transaction, its data verification and data synchronization and due to the cryptography secure, fast, global, reliable, correct, trustworthy, open-source, distributed, decentralized easy to use with new implementation combination by the use of hashing algorithm depends on proposed application for wallet/database/file system or for just cryptographic security.

V. FUTURE MERKLE TREE IMPLEMENTATIONS POSSIBILITIES

Considering the future implementation, always key concern is transaction charges to execute. For example, consumers frequently buy low-cost items using mobile wallet, such as Rs. 10 keychain, Rs. 100 mobile cover, Rs. 50 shampoo, etc. But, the bank involvement costs are very high and these costs are passed on to consumers as transaction fees or other charges. If we consider third-party payment then transaction cost increases and complete service provider cost must be paid. But the current system reversed, allow that transactions which are not paid with a service provider at risk.

The digital payment riot will have a huge impact on the digital and physical world by digital cryptocurrency. Proposed system must be decrease the transaction cost also manages until its successful execution completed in wireless environment. Here coin must be used vice-versa and authenticate by digital signatures Each mobile user transfers the coin to another user with digital signature hash of transaction and the public key to another mobile user and uses this coin for further transaction by double spending mechanism. Each user verifies the signatures before establishing a connection in wireless environment. Wireless connection can be Bluetooth, Radio Wave or NFC (Near Field Communication). After establishing a connection user can communicate securely and for more security user can also implement Merkle tree block chain for transactions such as used in Bitcoin. The proposed system payment transaction can work without the trusted third party due to the trusted payments system with the cryptographic proof using Merkle tree. It is implemented for peer-to-peer distributed network, which uses time-stamp server and the transaction's system generates time-based calculation proof to avoid the double payment.

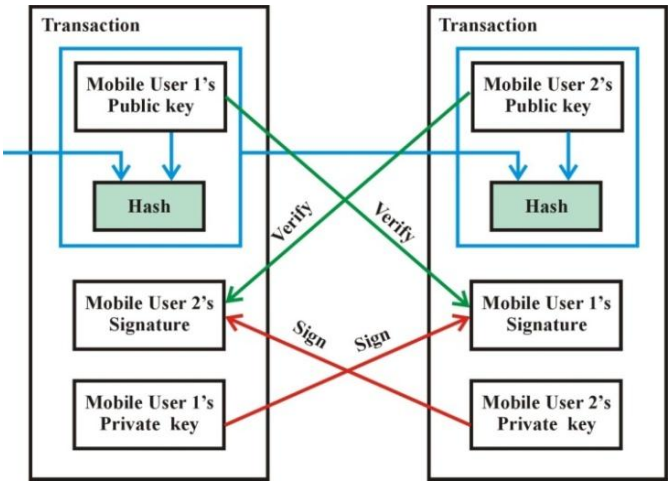

Figure 3. Communication between two mobile user using Merkle trees in wireless environment.

Proposed Merkle tree based system implementation by the combination of Graphical User Interface (GUI) based app with sensor detection techniques and NFC network. Here, overview of the individual implementations is as below:

[1]. According to US9858781B1 patent, GUI based mobile wallet app performs secures transactions using security server. Security server listens the user request and response a user's public key to the message. User's public key is mobile wallet user identification and access with transaction code. User's public key may be a matrix barcode, it can also allow to access user's multiple profiles. [2]

[2]. A mobile wallet type device contains a sensor and transceiver attached with the processor. To perform a secure transaction it follows blockchain smart contracts. The device can be accessed using digital key and a sensor monitor the data behavior with its access period

according to contractual ownership. The devices communication done based on sharing digital keys. A processor audits access rights and authenticate the contract using blockchain to identify total transaction cost. [3]

[3]. NFC technology is complement for various contactless communication technologies. NFC through many devices will be connected with each other. These NFC enabled device contains our preferences, health details, personal details and even our money details. These details affect to collect and exchange information, access control, health care, payment, reliability and coupons, transportations, and consumer electronics. So user no needs to carry any other identity cards and physical wallets anymore. NFC provides numerous benefits such as Intuitive, Versatile, Open and standards-based, Technology-enabling, inherently secure, Interoperable, Security-ready, etc. The blockchain technology is plays key role to decrease the cost of financial services by cost sharing through the mining. Thus financial institutions can reach out to the under banked and unbanked as well as those that need loaning and fund raising. These are possible through either decentralized or distributed peer-to-peer network of cryptocurrency. E-cash disrupted is solved in early 1990s. In future may sharing economy as well as assets of us by peer-to-peer network. Even rent the unused assets with fees. This can be done with NFC enabled devices efficiently without infrastructure change and digital trusts can be developed via blockchain technology. The NFC based devices, user has new way of life for medical care, education and finance services to monitor and improvement. [6]

VI. TECHNICAL REQUIREMENTS FOR MERKLE TREE IMPLEMENTATIONS

Knowing the technical requirements before implementations serves the basic precaution for optimal implementations such as the internal structure documentation is not exhaustive, hence reproducing the computation of the root hash would require some reverse engineering, the data structure resides fully in memory, and the root hash depends on the order of the updates. To implement this Proposed Merkle tree based system must follow the basic technical requirements are mentions below:

[1]. Merkle tree able to store, lookup;
[2]. Merkle tree able to compute root hash predictably by performing updates in a specific order;
- Helps in block header preparation and verification.

[3]. Merkle tree must have documented internal structure for reproducing the root has if needed to distinct implementation by application of updates in same order;
- Helps in block header verification.

[4]. Merkle tree able to return proof of existence in a verifiable format;
[5]. Merkle tree able to produce "proof of absence" for a key, i.e. a proof that no value exists in the tree for the key;
[6]. Persistence;
[7]. Space used only once for shared sub trees;
[8]. Garbage collection;
[9]. Erlang bindings.
[10]. Merkle tree whose root hash is dependent only on the data - not on the order of the updates;
- This may enable parallelization of application of some transactions in the block;
- This may ease block sharing.

[11]. Ability to persist nodes by direct key in the persistency layer, i.e. without requiring the persistency layer to generate an identifier at each insert;
[12]. Parameterized persistence;
[13]. Merkle tree data structure that is already established;
- This would reduce implementation effort in various languages;
- This may reduce documentation effort;
- An opportunity is using Ethereum's "modified Merkle Patricia tree" (a Merkle compact prefix tree) to utilize its features.

[14]. Performance.

VII. CONCLUSION

Merkle tree basically used for cryptocurrencies based wallets, file system with authentication, backup system, control system, database, protocol, cryptography, etc. Merkle tree has various implementation possibilities such as distributed network, protocol, file system, software but still new are pending to come to take advantage of it. Based on comparisons of Bitcoin, Ethereum, Hash-based Cryptography, Apache Cassandra, Btrfs, ZES, IPFS identify that Bitcoin is most secure wallet, Merkle signature scheme provides better authentication and Btrfs and IPFS are better than ZES file system. Also identifies that Merkle tree individually cannot be use for wallet security but it must used inside a blockchain. And just for Validation and Verification Merkle tree is most suited in the wallet. But this study analysis may differ according to the different implementation, combinations and functionality parameters. According to the application platform integration used for proposed system hashing algorithms may differ. Bitcoin is loses its popularity, a new crypto currency will emerge to substitute it with enhanced security features through the Merkle tree. Cryptographic security is not limited cryptocurrency but it can expand to connection and transaction through a Merkle tree. Each cryptocurrency is a great and an exciting experiment with Merkle tree for trust enhancement.

This paper gives the future new innovative implementation possibility by the combination of GUI, Sensor detection and NFC based innovative system with wireless environment for mobile wallet. Also state the basic

technical parameters to satisfy for the proposed system. In future we see a great rise of merkle tree use with NFC based mobile technology being the force behind its explosion. At that time, it is complicated to guess about cryptocurrency popularity in the world with lots of uncertainty. But the new technology comes with these combinations that cannot ignore by any financial institutions. New technology with merkle tree adapted for financial services, medical care, stocks and education. Combination of open-source technology and cryptocurrency data structure merkle tree may become an alternative of digital cryptocurrencies with merged to achieve new objectives.

ACKNOWLEDGEMENT

I would like to acknowledge the https://en.wikipedia.org for efficient definition content providing for this study. I thank my co-author Dr. Dharmendra Patel, Associate Professor, CMPICA, CHARUSAT, Changa for guidance, support and valuable comments to greatly improve the paper manuscript. Last but not the list, we offer our true regards to all of those who supported us in any respect during the completion of the paper.

REFERENCES

[1] Gencer, Adem Efe, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. "Decentralization in Bitcoin and Ethereum Networks." arXiv preprint arXiv:1801.03998 (2018).
[2] Campero, Richard, Sean Davis, Graeme Jarvis, and Terezinha Rumble. "Architecture for access management." U.S. Patent 9,858,781, issued January 2, 2018.
[3] Tran, Bao, and Ha Tran. "Smart device." U.S. Patent Application 15/807,138, filed March 22, 2018.
[4] Harm, Julien, Josh Obregon, and Josh Stubbendick. "Ethereum vs. Bitcoin." Creighton University, undated manuscript, retrieved 1 (2017).
[5] Ethereum community, "Ethereum Homestead Documentation" ,Release 0.1, March 01, 2017.
[6] Nian, Lam Pak, and David LEE Kuo Chuen. "Introduction to bitcoin." In Handbook of Digital Currency, pp. 5-30. 2015.
[7] Chebotko, Artem, Andrey Kashlev, and Shiyong Lu. "A big data modeling methodology for apache cassandra." In Big Data (BigData Congress), 2015 IEEE International Congress on, pp. 238-245. IEEE, 2015.
[8] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." white paper (2014).
[9] Benet, Juan. "IPFS-content addressed, versioned, P2P file system." arXiv preprint arXiv:1407.3561 (2014).
[10] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
[11] Hülsing, Andreas, Stefan-Lukas Gazdag, Denis Butin, and Johannes Buchmann. "Hash-based Signatures: An outline for a new standard."
[12] https://en.wikipedia.org/wiki/Merkle_tree
[13] https://www.investopedia.com/terms/m/Merkle-tree.asp
[14] https://en.wikipedia.org/wiki/Hash-based_cryptography
[15] https://en.wikipedia.org/wiki/Apache_Cassandra

Authors Profile

Mansi P. Bosamia, B.C.A., M.C.A. and Pursuing Ph.D. at Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Charotar University of Science and Technology, Changa, Anand, India. She has more than 5 years teaching experience and 3 years Research experience. She has published/presented 6 papers in national/international conferences/journals of repute. She wrote 2 books related Data Structures and Algorithms. Her areas of interest are Computer Algorithms, Data Structures, Networking, Computer Graphics, Mobile Computing, Cryptography, etc.

Dr. Dharmendra Patel, B Sc.(I/C), MCA, SET, Ph.D. (Computer Science). He is working as Associate Professor at Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Charotar University of Science and Technology, Changa, Anand, India. He has published/ presented more than 20 research papers in national/ international journals/conferences of repute. Also, served as editorial/review board members in many international journals and have reviewed 25 papers of different international journals of repute. His areas of interest are Web Mining, Distributed Operating Systems, Cloud Computing, Soft Computing, Software Engineering, Data Structures, and Mobile Computing.

