Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 2, February : 2024

BLOCKCHAIN BASED SECURITY IN CLOUD COMPUTING ARCHITECTURE

Subhasini Pallikonda, Kalluri Sai Gayathri, Sama Manasa, Kallem Diksha Bhavani Reddy, Juluru Akshay, Imranuddin Shaik, Computer Science and Engineering Maturi Venkata Subba Rao Engineering College (Osmani Univeristy Affiliation) Hyderabad, India.
subhashini_cse@mvsrec.edu.in

***Abstract*— Cloud Computing pose security risk to the user's information. The principal drawback of Cloud Computing is the security troubles in storing data in the Cloud. Data manipulation is a serious threat to data integrity that could arise in cloud computing. Cloud computing customers want to be confident that their data is accurate and trustworthy. On other side, blockchain is a tamper-proof virtual ledger that can be used along with cloud to offer a tamper-proof cloud computing environment. In this paper, we suggest a scheme that combines both, cloud computing and blockchain technologies that guarantees data integrity for all homomorphic encryption schemes. The proposed scheme relies on Byzantine Fault Tolerance to overcome the cloud service provider's (CSP) ultimate authority over the data by building a distributed network of Cloud Service Providers (CSP).**

Keywords—Cloud Computing, blockchain, byzantine fault tolerance (BFT), homomorphic encryption (HE), CSP (Cloud Service Provider)

I. INTRODUCTION

Cloud computing has been implemented to numerous IT environments because of its availability and efficiency. Additionally, cloud security and privacy issues had been discussed in terms of important security factors: integrity, confidentiality, access control, authentication, and so on. Data security is mostly characterised by means of data protection threats. The cloud computing domain is also prone to numerous threats. The primary reason for this is that cloud computing combines many unique technologies in its operation. It is paramount to apply the risk management process to equalize the benefits of security risks, and cloud computing[1].

Using blockchain can provide higher protection as compared to storing all data in a central database. In the data storage and management aspect, damage from attacks on a database may be avoided. Moreover, blockchain can provide transparency in data when carried out to a place requiring the disclosure of information because the blockchain has an openness attribute. Due to such strengths, it may be utilized in numerous areas such as the financial zone and the Cloud Computing environment and its applications are predicted to grow.

To mitigate the risks related to cloud computing CSA (Cloud Security Alliance)[2] has laid out critical shared duties for cloud service providers and the customers. The activity of the CSP is to document, design and put into effect the client safety control and internal security control. Despite the CSP's tries to establish a sturdy security base, such preparations are rarely substantive from the data owners' view, mainly when it comes to trusting the CSP itself. This is compounded by the fact that the growth of cloud computing technology leads to new security vulnerabilities and amplifies existing ones. This report provides the Blockchain Technology integration with the Cloud Computing Architecture to provide security for the users' data.

II. PRELIMINARIES

A. Blockchain Technology

Implementing BC techniques in cloud scenarios have attracted considerable attention in both academia and industry. BC technology, in essence, consists of distributed digital blocks bound to each other based on cryptographic principles. Each block contains a cryptographic hash of the previous block, a timestamp and transaction data. BC grant all participants the ability to authenticate

transactions independently on a peer-to-peer network. To approve and record transactions in the BC, a consensus mechanism is required to ensure that the network of nodes is in agreement. Once a block is validated, it cannot be altered retroactively without modification of all subsequent blocks. [3]NIST defines BC technology as follows: (Blockchain): Distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance). New blocks are replicated across copies of the ledger within the network, and any conflicts are resolved automatically using established rules. Several businesses catering to the interest in BC technologies by developing cloud-based BCs.

Well-known CSPs have provided Blockchain as a Service (BaaS) to their clients based on the Software as a Service (SaaS) model. Launched the []Amazon Managed Blockchain using open-source software platforms such as Ethereum and Hyperledger Fabric that allows developers to create and share information in a decentralised manner easily. The introduction of Bitcoin did the introduction of Blockchain technology. Bitcoin is a form of digital currency introduced by a pseudo name called ‘‘Satoshi Nakamoto’’ in 2008. He published a white paper, ‘‘Bitcoin: A Peer to Peer Electronic Cash System,’’ which presents us with the direct online payment from one party to another without using any third party.

This electronic cash system mainly overcomes the problem of double-spending the money, primarily the digital currency nature that allows being easily duplicated and spent more than once. This problem is solved by linking each transaction with one another in a tamper- resistant manner. The public ledger is being used to connect transactions in a tamper- resistant way. With this ledger, a network can verify the transaction history that the user submits for payment and can confirm that the coin has not already been spent. In comparing

The blockchain is an indestructible digital ledger for keeping track of economic transactions that can be programmed to maintain not only financial transactions but virtually everything that has a value. When we implement blockchain technology, no government interference is needed, and zero percent of fraud due to consensus validation. By eliminating the involvement of third-party, instant transactions can be done without paying transaction fees. These features improve financial efficiency.

B. Cloud Computing

Cloud computing is an internet-based computing technology, where shared resources such as software, platform, storage and information are provided to customers on demand. Cloud computing is a computing platform for sharing resources that include infrastructures, software, applications, and business processes. Cloud Computing is a virtual pool of computing resources. It provides computing resources in the pool for users through internet. Cloud computing, as an emerging computing paradigm aiming to share storage, computation, and services transparently among a massive users. The exact definition of cloud computing is A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.

The deployment models for the cloud are as follows:

- Public Cloud – the cloud infrastructure is available for public use. The cloud infrastructure and resources are owned and managed by an organization dealing with cloud services.
- Private Cloud – the cloud infrastructure is specifically for the use of an organization. The cloud infrastructure and resources can be owned and managed by this very organization or a third party.
- Hybrid Cloud – is a combination of public and private cloud. Generically, organizations are outsourcing public clouds for functions which are not confidential or which may not come under scanner.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 2, February : 2024

The cloud offers the following service models:

- Infrastructure as a Service (IaaS) – refers to a service model in which infrastructure is lent out to the cloud users. IT resources such as servers, storage, operating systems, network devices, etc. are provided to the users for hiring. These resources are placed onto the cloud so that the users can avail their services as per their requirements.
- Platform as a Service (PaaS) – refers to a service model in which the application development toolkit is lent out. The cloud users hire these services to develop and deploy their own applications as per their own requirements. Here the user is not required to own these deployment tools and development environment.
- Software as a Service (SaaS) – refers to a service model in which the applications are lent to the cloud user for his requirements and specifications. The examples of such applications include Sales Management, Finance and Accounting and Payroll Systems amongst others.

III. TECHNIQUES AND ALGORUTHMS USED

The proposed approach in this paper adopts BFT, HE and Hash Function in a unified approach for maintaining data confidentiality in cloud computing. Important concepts from these are detailed in Subsections III-A, III-B and III-C respectively.

A. Byzantine Fault Tolerance (BFT)

Practical Byzantine Fault Tolerance is a system that has a primary node and secondary nodes. These nodes work together to reach a consensus, making this system one of the solutions to the Byzantine Generals Problem.

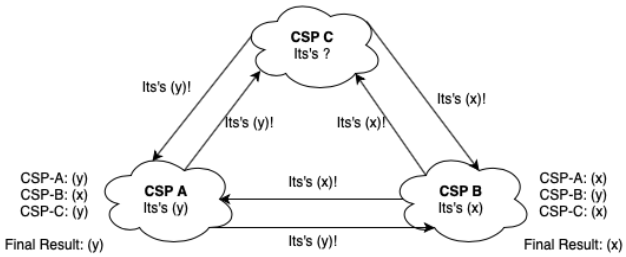

Fig. 1. BFT Concept

Here's a basic breakdown of how practical Byzantine Fault Tolerance works:

- The client makes a request to the primary node.
- The primary node sends that request on to the secondary nodes.
- The nodes process the request, provide the service, and respond to the client.
- The client waits until it has received the same response from m+1 nodes, with m being the maximum number of faulty/malicious nodes the system allows.

In a practical Byzantine Fault Tolerance system, the maximum number of faulty/malicious nodes can't be equal to or greater than one-third of the system's total nodes.

B. Homomorphic Encryption (HE)

Homomorphic Encryption is a form of encryption that lets in users to carry out computations on its encrypted records without first decrypting it. Thus ensuing computations are left in an encrypted form which, whilst decrypted, bring about an equal output to that produced had the operations been performed on the unencrypted statistics. Homomorphic encryption can be used for privateness-keeping outsourced storage and computation. This permits data to be encrypted and out-sourced to industrial cloud environments for processing, all whilst encrypted. For sensitive facts, consisting of fitness care statistics, homomorphic encryption may be used to allow new offerings with the aid of doing away with privateness obstacles inhibiting facts sharing or growth safety to existing offerings. For example, predictive analytics in health care can be difficult to use via a 3rd party carrier issuer due to medical records privateness issues, but if the predictive analytics service provider can operate on encrypted information instead, these privacy issues are diminished.

The conversion of data into ciphertext in which the system has the capacity to behaviour operations on information that is encrypted without any reach to the private decryption key; the proprietor of the records have to be the handiest one in possession of the private key. In the procedure of making use of mathematics operations to encrypted data, the equal outcomes ought to be gotten as inside the case of unprocessed statistics.[5]-[7]

To sum up, HE has four main operations, namely: *KeyGen*, *Enc*, *Eval* , *Dec* :

- The first process of HE is key generation *KeyGen*, where the data owner creates the public-key pair (a public key *puk* and a private key *prk*).
- The next is the encryption process *Enc* which involves applying the encryption algorithm onto the data $C = Enc_{puk}(P)$ before sending it to the cloud server.
- The evaluation process, Eval. In this, the cloud server performs the requested calculation on the encrypted data before sending the result back to the client in its encrypted form.
- With the corresponding *prk* , the client is able to process the decryption function, *Dec*, to recover the plain- text.

C. Blockchain Hash Function

A hash characteristic takes an input string (numbers, alphabets, media documents) of any length and transforms it into a set period. The fixed bit length can vary (like 32-bit or 64-bit or 128-bit or 256-bit) depending on the hash feature that's getting used. The fixed- duration output is called a hash. This hash is likewise the cryptographic spinoff of a hash algorithm. We can apprehend it from the Fig. 2.

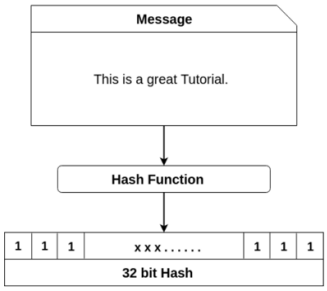

Fig. 2. Hash Function

The hash algorithm has certain unique properties: It produces a unique output (or hash). It is a one-way function. In the context of cryptocurrencies like Bitcoin , the blockchain uses this cryptographic hash function's properties in its consensus mechanism. A cryptographic hash is a digest or digital fingerprints of a certain amount of data. In cryptographic hash functions, the transactions are taken as an input and run through a hashing algorithm which gives an output of a fixed size.

Since the Hash function is a one-way function, there is no way to get back entire text from the generated hash. This is different from traditional cryptographic functions like encryption where you can encrypt something using the key and by using decryption, you can decrypt the message to its original form.

IV. IMPLEMENTATION

The verified computation design is based on CSP and BC technologies which both play an equally crucial role. The proposed scheme basically consists of four phases which are Encryption Phase, Outsourcing/Uploading Phase, Chaining Transaction and Verification Phase. There are three main components which are required throughout this phases whose roles are:

1) *Client:* To perform verification by comparing master hash values from every CSP based on the received block header information.

2) *Multi-CSPs:* A client will hire more than one CSP. Each CSP has its very own agreement with the client but all are subjected to the same terms. The *n* number of hired CSPs will carry out computations, which will produce a master hash for their data and forward the result to the BC-based application.

3) *Blockchain:* Creates new blocks that contain the master hashes as a transaction, then returns the block header to CSP.

Before going through the process, the client needs to decide two main aspects for the workflow of the design: the frequency of computing master hash values (decided with a frequency variable, *t*) and the corresponding cryptocurrency wallet. *t* determines the number of computations requested by a client before the multi-CSPs compute the master hash of their corresponding databases. The value of *t* depends on two primary factors. The first is the client's data growth percent, and the second one is his financial ability to pay the BC transaction charges.

A. Encryption Phase

In this phase the client's data is encrypted using the homomorphic encryption algorithm. First the public key and private key are generated. Then the HE is applied on the data using the public key. Now, if the user requests for some calculations then they are performed by the CSP on the encrypted data. The HE Data is then provided to the next phase.

The use of HE schemes alone does not guarantee full data security. Data integrity can still be compromised by CSP and can go undetected. Taken a scenario the CSP can substitute a
cipher text with some other valid ciphertexts without knowing the content of the substituted data. Therefore, data integrity needs to be enforced on such outsourced computations, which can be achieved by the decentralized database.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 2, February : 2024

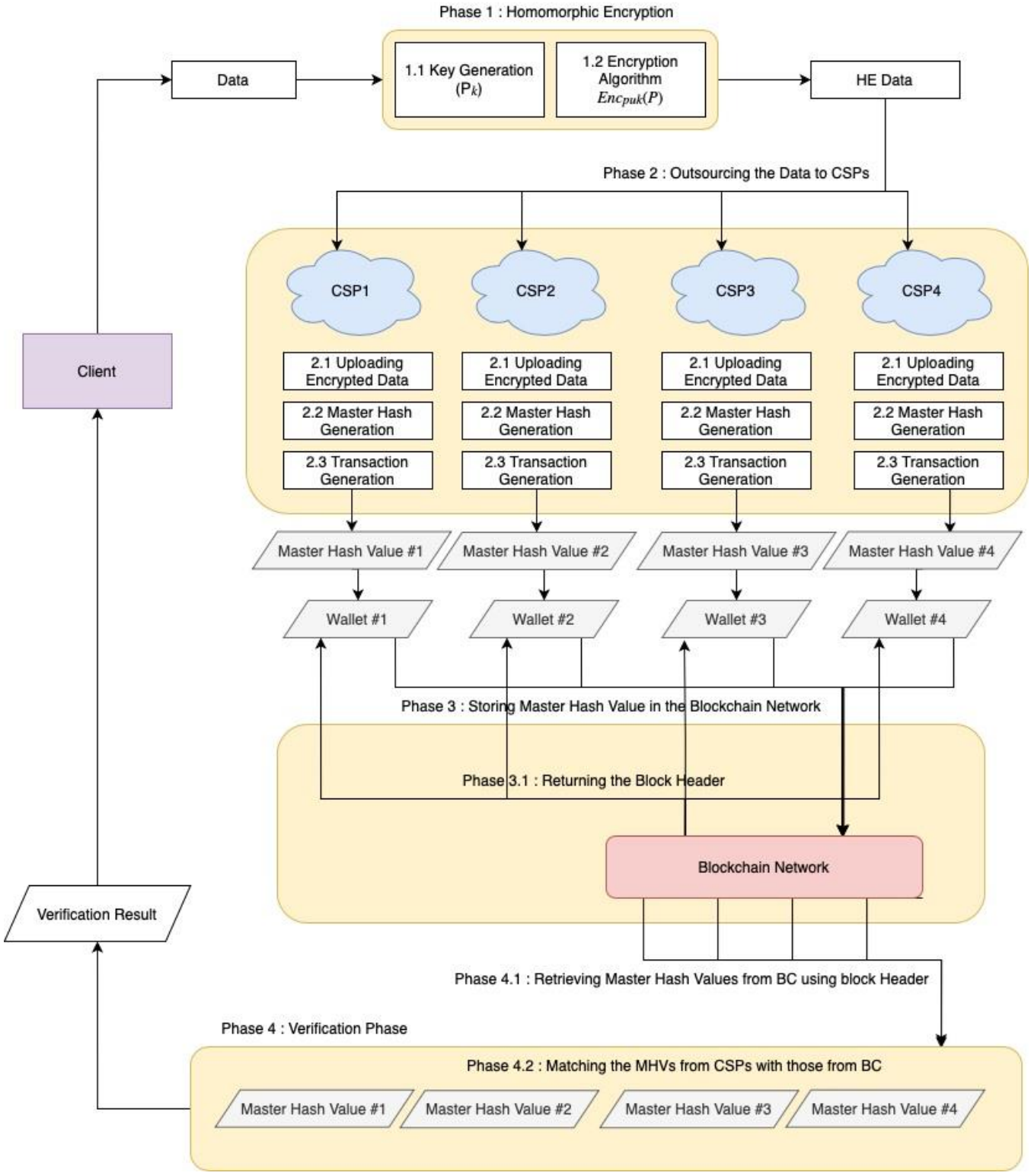

Fig. 3. Flow Diagram

B. Outsourcing HE Data and Master Hash Generation

In this scheme to carry out verification within the CSP environment, we adopt certain BC's BFT consensus features and put them into practice. It may even rely upon hash functions as well as the properties of the distributed ledger. The number of hired CSPs is determined based on the BFT scenario. If f CSPs are Byzantine (or malicious), and the system consists of $2f + 1$ CSPs, the malicious CSPs coordinate to say arbitrary things to the other $f + 1$ nodes. Thus, the upper limit of f for

Byzantine faults should be set to $f < N/3$. Therefore, in order to tolerate one Byzantine node, the minimum requirement is N = 4 CSPs, which we considered in our scheme.

The homomorphic encrypted (HE) data is now outsourced to the multi-CSPs, where it is stored in the database. Now the CSPs. The operations performed by each CSP are:

1) Calculate the Master Hash Value (MHV) of its database by applying SHA-2 after *t* timesof requested computations.
2) The CSP then saves this master hash in a transaction log, to be stored in the BC network

C. Blockchain – Storing Master Hash Value

To store the master hash values, we can consider two popular cryptocurrencies, Bitcoin and Ethereum. But in our case we would be considering Ethereum blockchain. There are two possible master hash storage cases for the Ethereum account type.

1) *Externally Owned Accounts (EOA):* Normal Ethereum wallets store private keys and offer a public ETH address for user accounts [8]. To perform an ETH transaction, each of the hired CSPs has to get a normal wallet. Now prepare the transaction and embed master hash value. The transaction is released in the ETH network only after it is signed by the initiator, which then validated in the network.
2) *Smart Contract Accounts:* A multisig wallet in Ethereum is a smart contract deployed for storing ETH that belongs to multiple owners. Each transaction must be approved by a specified number of owners [9]. The client will deploy four shared smart wallets, one for each CSP. The CSP and two client accounts are the owners for each wallet, to follow 2/3rd majority rule. Smart contract deployment in the Ethereum network is performed via transactions. The transaction structure is the same as an EOA, but the data included in the transaction differs as the input data should include the bytecode plus any encoded arguments if required by a constructor [10]. A transaction_Id or hash code is returned to CSP as a response. The CSP will disclose this transaction_Id to clients so that the clients can validate it.

D. Client Phase – Verification

After master hash is stored in the blockchain, it is the duty of the client to verify that the values sent by all CSPs are identical. We would be seeing the verification in the ETH blockchain as considered network.

Verification using Ethereum will vary relying on the sort of account getting used. For EOAs, every CSP is needed to send the block header to the client for each transaction. This permits the consumer to track and perform verification of all transaction. As for CAs, the client can get the block headers associated with the CSPs' transactions from the multisig wallets. Thus, the verification method may be executed based at the timestamp data in each block header. The approach primarily based on CA outperforms EOA due to its capacity to set up the shared wallet, which allows the data verification process.

V. RESULT ANALYSIS

The verified computation design is based on CSP and BC technologies which both play an equally crucial role. The proposed scheme basically consists of four phases which are Encryption Phase, Outsourcing/Uploading Phase, Chaining Transaction and Verification Phase. There are three main components which are required throughout this phases whose roles are:

A. Cost and Performance Analysis

Ethereum cryptocurrency trading is 97% lower than the Bitcoin. 1 ETH costs approximately $1,435.15 USD[1]. The cost every zero-byte transaction is at least 21,000 gas. Each additional byte incurs a cost of 68 gas. Thus, the total cost of storing the master hash value generated from SHA-2 is about $0.006 USD. The estimated cost of transaction fees if transactions are performed continuously every half an hour is around $105 USD per year. Table 1 provides a quantitative comparison of verification overhead costs for embedding data in Ethereum

Embedding master hash values in an Ethereum transaction will not require opcodes. In contrast, Smart Contract (CA) option will require many function calls to store the master hash. Table 2 illustrates the comparison between cost and performance options.

TABLE I. OVERHEAD COST ANALYSIS

A) EOA COSTS

Transaction Fee Total cost in	**Master Hash every**			
	30 min	***One hour***	***Half hour***	***One day***
One day	$0.288	$0.144	$0.012	$0.006
One month	$8.76	$4.38	$0.365	$0.1825
One Year	$105.1	$52.56	$4.38	$2.19

B) SMART CONTRACT (CA) COSTS

Transaction Fee Total cost in	**Master Hash every**			
	30 min	***One hour***	***Half hour***	***One day***
One day	$0.28	$0.12	$0.01	$0.005
One month	$7.3	$3.65	$0.304	$0.152
One Year	$87.6	$43.8	$3.65	$1.825

TABLE II. OVERHEAD COST VS PERFORMANCE COMPARISION

Option	**Performance**	**Cost**
EOA	3 ★	4 ★
Smart Conract (CA)	4 ★	1 ★

1: Least Favorable, 2: Less Favorable, 3: More Favorable, 4: Most Favorable

[1]As of 18 September 2022

B. Security Analysis

Security is analysed in terms of confidentiality, privacy and data integrity.

Data Confidentiality: This is achieved using the Homomorphic Encryption to encrypt the data before storing it in the cloud.

Privacy: The client can authorise a CSP to perform data processing via the HE scheme, by providing the public key of the encrypted data to the CSP.

Data Integrity: Using the concept of Byzantine Fault Tolerance the master hash value generated by CSPs is stored in the blockchain. The block header will be provided to clients for verification purposes.

VI. CONCLUSION

This paper focuses on the addressing the data breach in the cloud computing by the cloud service provider's authority over the client data. We have approached this issue in this paper using the homomorphic encryption to provide data confidentiality and privacy. While the data integrity is ensured by the approach of distributed network of CSPs and Byzantine Fault Tolerance consensus of

the Blockchain Technology. We have implemented this approach by simulating a local Ethereum Blockchain using Truffle and Cloud Service Providers. The CSPs need to generate the master hash value of their database and store it in the blockchain. These values can be used by the clients for the verification of their data. We have also analysed the cost and performance measures along with the security requirements.

Although this scheme has various advantages, it cannot provide information about which data records have been attacked or tampered with. Hence, in future we aim to implement this feature that can pinpoint exactly where the data has been compromised in the CSPs.

REFERENCES

[1] Parul Chachra, A Survey of the Existing Security Issues in Cloud Computing, IJCSIT, Vol.5(2), 2014

[2] Cloud Security Alliance. (2017). *Security Guidance V4.0.* [Online]. Available:https://cloudsecurityalliance.org/download/security-guidance-v4/

[3] D. Yaga, P. Mell, N. Roby, and K. Scarfone, ‘‘Blockchain technol- ogy overview,’’ 2019, arXiv:1906.11078. [Online]. Available: http://arxiv. org/abs/1906.11078

[4] Amazon Web Services. (2020). *Announcing General Availability of Amazon Managed Blockchain*. [Online]. Available: https://aws.amazon.com/ar/about-aws/whatsnew=2019=04/ introducing-amazon-managed-blockchain/#::text=Amazon%20Web% 20Services%20%28AWS%29%20announces,Hyperledger%20Fabric%20is%20available%20today

[5] E. Orsini, N. P. Smart, and F. Vercauteren, ‘‘Overdrive2k: Efficient secure MPC over Z2k from somewhat homomorphic encryption,’’ in *Proc. Cryptographers’ Track RSA Conf.* Cham, Switzerland: Springer, 2019, pp. 254–283.

[6] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, ‘‘A survey on homomorphic encryption schemes: Theory and implementation,’’ *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, Sep. 2018.

[7] T. S. Fun and A. Samsudin, ‘‘A survey of homomorphic encryption for outsourced big data computation,’’ *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 8, pp. 3826–3851, 2016.

[8] *Ethereum Wallets*. Accessed: Sep. 19, 2020. [Online]. Available: https://Ethereum.org/en/wallets/

[9] M. di Angelo and G. Salzer, ‘‘Wallet contracts on Ethereum—Identification, types, usage, and profiles,’’ 2020, arXiv:2001.06909. [Online]. Available: http://arxiv.org/abs/2001.06909.

[10] S. Rouhani and R. Deters, ‘‘Performance analysis of ethereum transactions in private blockchain,’’ in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2017, pp. 70–74.