Semiconductor Optoelectronics
ISSN: 1001-5868

BLOCKCHAIN BASED SECURE PERCEPTRON NEURAL BONEH LYNN SHACHAM AND SCHULZE VOTING FOR CLOUD COMPUTING ENVIRONMENT

[1]P.M. Pazhani Selvam
Reg. No: 18123152161002, Research Scholar, Computer Application, Manonmaniam Sundaranar University, Tirunelveli-627012

[2]Dr. S.S. Sujatha
Associate professor, Dept. of M.C.A., S.T. Hindu College, Nagercoil-629001

[3]Dr. K.K. Thanammal
Associate professor, Dept. of M.C.A., S.T. Hindu College, Nagercoil-629001
Email: [1]pmpselvam69@gmail.com, [2]sujaajai@gmail.com, [3]thanaravindran@gmail.com

Abstract

In today's world, the transaction through the cloud computing environment has experienced considerable awareness and constructs several applications that can significantly operates large amounts of records on an increasing demand globally. As a consequence of these enormous developments, the uneasiness elevates in terms of security threats in the cloud computing environment. Thus, we need to identify a robust solution that can maintain the confidentiality and authenticity of data while ensuring appropriate services. Recently, blockchain technology handles these issues on numerous platforms via Machine Learning techniques to share transaction records by storing the immutably. However, the traditional blockchain architecture gives rise to numerous issues as a consequence of constrained potentialities, therefore compromising transaction latency and communication cost. Also with the increasing size of blockchain, data flow also increases and also affecting data confidentiality. Hence, to overcome these challenges, in this work, a novel method called, Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) secured data communication in cloud computing environment is proposed. In the BBLSV-PN method, the registration of a cloud data owner is performed to enable legitimate access control thus ensuring traceability. Moreover, the communications take place in the form of transactions of blockchain via Cloud Server in cloud computing environment. We propose a lightweight consensus mechanism using Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation to ensure that each cloud data owner is in control of its block. The proposed method also handles the secure data communication between cloud data owner and cloud user in cloud computing environment by employing Perceptron Neural Network-based secure communication model. The proposed method has been validated in a simulated CloudSim environment and the results are promising in terms of numerous metrics. Numerical validations have also been provided in context of transaction latency, communication cost and data confidentiality rate.

Keywords: Cloud computing, Blockchain, Machine Learning, Boneh–Lynn–Shacham, Schulze Voting Consensus Perceptron, Neural Network

1. Introduction

Cloud computing environment stores and accesses data via a single sever. Cloud computing environment bestows different types of services and hence several organizations have started utilizing this technique. The two main advantages in cloud computing environment are its user friendly nature and the cost effectiveness. This remains some of the significant advantages for storing and accessing the data. In addition, security remains one of the major factors in cloud computing environment for cloud users. Blockchain is an appropriate and decentralized framework that can augment trust in the derivation or provisioning of data in the cloud computing environment.

A novel method called, Blockchain and Smart Contrast-based Data Provenance (BSCDP) was proposed in [1] for ensuring secure storage in the cloud computing environment. Initially, biometrics obtained via fingerprint and uncloneable function via physical factors were utilized in the verification process, that in turn ensured secure data transmission with minimum verification time. Next, to ensure both security and privacy fuzzy extractor was utilized along with the elliptic-curve key based transposition cryptography mechanism. Finally, blockchain and the inter planetary file system (IPFS) were introduced with the purpose of reducing the computational overhead involved in storing. Finally, data verification was performed via provenance auditor that in turn reduced the response time and decryption time considerably.

A Homomorphic Block Ring Security System (HBRSS) was proposed in [2] with the objective of ensuring high security during the process involved in transmitting data and also encrypted the data in a quick and safe manner via third party. Initially, the encrypted data was split into a distinct numbers of blocks. These blocks were then stored and associated in the form of a ring. Moreover during data transmission, the data was also said to be safeguarded against eavesdropping and tampering attacks from unauthorized users. In addition during the data processing stage, block-ring was constructed in a safe and effective fashion. As a result, the CPU utilization was reduced with minimum execution and encryption time.

A blockchain - based secure data processing method was proposed in [3] that included the formulation of multi-objective optimization problem. Moreover, an optimal container-based data processing method and a blockchain-based data integrity management method were also designed with the objective of reducing link breakage and latency.

With the explosive nature of the internet era a major transformation has erupted in data storage and accessing. One such new trend is the cloud computing environment. A computationally secure key generation mechanism was proposed in [4] to safeguarding the data via encryption. Initially, an encryption model based on SVM was first designed and then to make the procedure more complicated optimization techniques were considered that in turn ensured computationally efficient and secure mechanism for Cloud environment. Yet another blockchain based cloud infrastructure for internet of things was introduced in [5].

A blockchain-based decentralized distributed storage and sharing mechanism to entrust end-to-end encryption and fine-grained access control was presented in [6]. Here, in the auditable access control layer, access control mechanism was designed based on the attribute function by employing Ethereum blockchain. Also smart contracts were levied via interplanetary file system (IPFS). With this not only the transaction cost was improved but also resulted in the improvement of system throughput.

Over the past few years, distinct numbers of protocols were presented for Remote data integrity checking (RDIC). Moreover, Identity (ID) based RDIC protocols were also constructed that in turn not only guaranteed integrity of data but also ensured privacy to cloud data. This was constructed with the assumption that the RDIC were found to be full proof but however not so in real world applications.

In [7], a novel methodology of RDIC with un-trusted PKG and in the presence of malicious CS, a partial key along with the three distinct services, namely, authentication, authorization and accounting was designed. With this three distinct services the time involved in secured key generation were found to be reduced significantly. A survey covering in depth analysis of probable data privacy and thread mechanisms were investigated in [8]. A novel privacy mechanism concentric towards data owner in data-owner centric privacy model in cloud environment was proposed in [9]. The method integrated two significant paradigms for ensuring privacy of cloud data, namely, Attribute-Based Encryption (ABE) and blockchain. With these two mechanisms, the data owners privacy protection were said to be strengthened.

This paper extends our previous work [1] and [2] by concentrating on communication cost and transaction latency, by broadening the consideration by taking the additional use case into account as well as by conducting a more rigorous evaluation of data confidentiality rate of the solution, and by providing a discussion about probable solutions to ensure secure data communication between cloud entities via blockchain and machine learning. Moreover, the related work has been improved and broadened.

1.1 Contributing remarks

The contributions of this work are as follows:

- To propose a method called, Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) secured data communication in cloud computing environment.
- We establish a cloud computing network model using cloud computing and blockchain for secure communication between cloud entities.
- The proposed method employs cloud computing for efficient data sharing, and it uses blockchain for data verifiability and confidentiality via Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation.
- We propose secure communication model and key agreement techniques between cloud data owner and cloud user via cloud server to ensure secure communication in public channels by employing Perceptron Neural Network-based secure communication model. The proposed method can guarantee numerous security features including data confidentiality.

- We formally analyze and show the robustness of the proposed method. Furthermore, a detailed comparison of the proposed method with other recently proposed methods are provided to demonstrate its efficiency and security.

1.2 **Structure of the paper**

In Section 2, related work on blockchain, machine learning and cloud computing environment for secure communication is introduced. In Section 3 the proposed cloud computing network model is presented and proposes a Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) in cloud computing environment. In Section 4, the proposed method is analyzed by comparing with other related methods in Section 5, we conclude the paper.

2. **Related Works**

The acceptance in the cloud application materializes several tools and techniques to the users that in turn make as a hot research area to numerous researchers for past few years. Also the low cost and ease in utilizing the applications that in turn has high accuracy dispenses a freedom to the users upon comparison with the other prevailing generation applications.

A secure data transmission technique using hybrid meerkat clan algorithm via distributed cloud server was designed in [10]. Moreover, a lempel–ziv algorithm based on a two-stage model to ensure security was also presented. However with the improvement in technology security breaches are also found to be increasing owing to the load found in the increased numbers of users. A secure cloud computing mobile framework that ensures meritorious service to users employing hybrid Rivet–Shammir–Adleman algorithm and Elliptic Curve Cryptography was presented in [11]. With this hybrid model not only the memory utilization was improved but also the computation time involved was reduced significantly.

In [12], an enhanced security environment was designed with the objective of securing the cloud users data in the cloud computing environment. Initially, a key generation algorithm employing Elliptic Curve Cryptography was first designed for obtaining secured keys. Second, an access control mechanism based on elliptic curve was designed that in turn ensured data accessibility. Finally, a digital signature algorithm based on modulo function was presented that in turn proved to be not only highly integrity but also ensured security.

A systematic review and analysis on secured data storage and access mechanism was investigated in [13]. Taxonomy of blockchain and machine learning techniques for secure data communication in cloud computing environment were presented in [14]. Yet another survey of data security and privacy protection mechanism for cloud storage was investigated in [15]. In [16], some of the significant issues faced by the cloud and some solutions by combining it with blockchain were investigated. Also a brief investigation concentrating on blockchain combining with the cloud was also presented. Moreover, a detailed architecture combining and communication between blockchain and cloud was also designed. Deep learning techniques for secure communication via blockchain were introduced in [17].

A three phase model for ensuring authentication based security was proposed in [18]. Here, first, a star structure was designed that created keys for within cluster. Intra cluster communication was focused in the second phase. Finally, authentication protocol was designed for ensuring security. With this three phase mechanism not only end to end delay was reduced but also resulted in the improvement of packet delivery rate. An optimized blowfish algorithm was employed in [19] based on multi-stage authentication mechanism that in turn guaranteed secure data retrieval. A secure transmission model combining comprehensive sensing based cipher and edge computing was presented in [20] for ensuring privacy preservation.

Different from the previous works, the proposed Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) secured data communication is an integrated method utilizing blockchain and machine learning model in cloud computing environment. Our method with the machine learning aspires to discern secure data communication.

3. Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) secured data communication

The proposed Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) in cloud computing environment design is a distributed secure framework to protect the cloud data owners from threats. The BBLSV-PN method combines, blockchain, Machine Learning and cloud computing to attain data confidentiality with minimum transaction latency and communication cost. Every cloud data owner and cloud user necessitates in seeking permission from cloud server to be authenticated. This permission is said to be examined during each block entering into the cloud computing network. In order to ensure authentication, the cloud data owner and cloud user needs to register to have a share in sending blocks in corresponding transactions. Therefore the three steps involved in the designing of the proposed method are cloud data owner registration, block generation and validation and secure communication. First, with the designed cloud computing networking model cloud data owner registration process is done. Second for each registered cloud data owners, Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation model is designed. Finally, a secure communication process is designed by means of Perceptron Neural Network-based secure communication algorithm. The detailed description of Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) in cloud computing environment is elaborated in the following sections.

3.1 **Cloud Computing Network model**

In this section, cloud computing network model using blockchain is presented. The proposed method consists of four entities. They are, cloud server 'CS', cloud data owner 'DO', cloud data user 'DU' and blockchain 'B' respectively. A comprehensive illustration of each entity in diagrammatical representation is given below in figure 1.

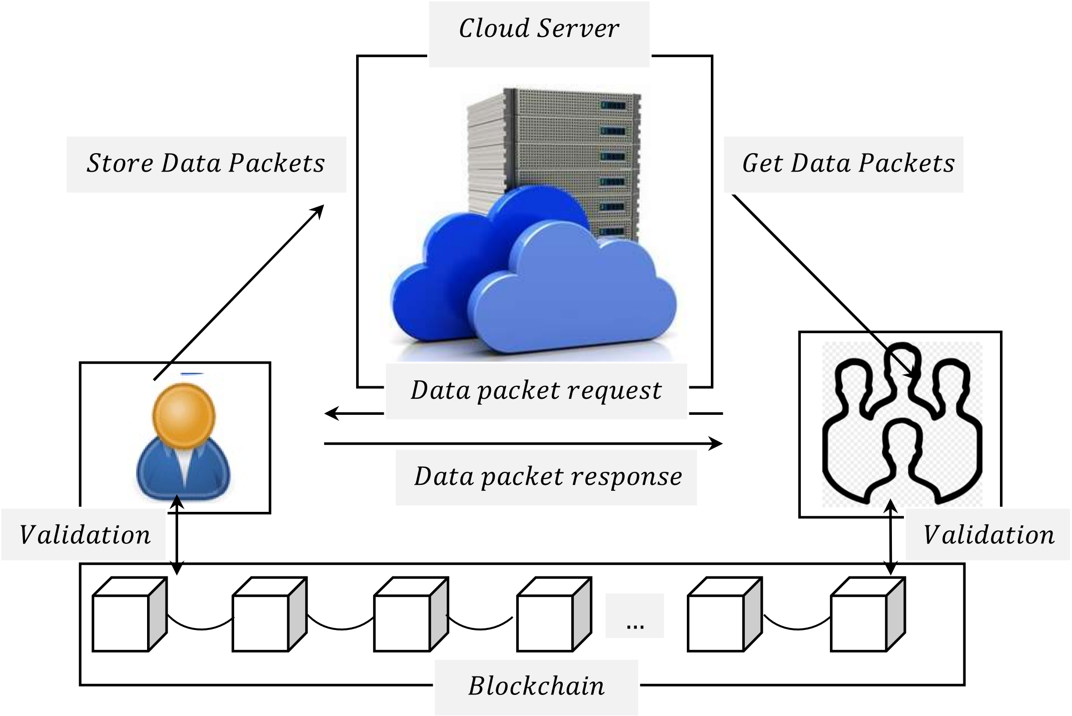


Figure 1 Cloud Computing Network model

As shown in the above figure, the cloud server 'CS' has adequate computation potentiality to replicate the data packets 'DP' acquired and in turn stores the generated 'DP'. The 'CS' acquires the 'DP' from cloud data owners 'DO' upon successful accomplishment of mutual l key agreement. Then the cloud server share data packets with cloud data user after acquiring cloud data owner confirmation. Moreover, the 'CS' also uploads hash values of the stored 'DP' and log 'DP' to the blockchain. Next, the cloud data owner 'DO'send the data packets 'DP' to the cloud server 'CS' upon successful mutual authentication. In addition, when a 'DO' receives a 'DP' request from a 'DU', the 'DO' authorizes the cloud to share 'DP' with the cloud data user. Once 'DP' sharing is done, the 'DO' checks the log record of the 'DP' via the blockchain.

Third, the cloud data user 'DU' request data packets 'DP' as required. Once the data packets 'DP' are mutually authenticated with a 'DO', the 'DU' have the access of the can access the digital data of the data owner stored on the cloud server. Followed by which, the 'DU' can acquire the necessitated 'DP' and verify the data integrity via the blockchain. Finally, the blockchain 'B' stores hash values of the 'DP' stored on the 'CS' and log records between the 'CS' and 'DU'. The data hash values are utilized by 'DU' to verify that the 'DP' are not altered, and the log records are utilized by 'DO' to inspect that their 'DP' are shared with permissible 'DU'. Each transaction comprises of a signature and is uploaded only upon successful verification of the signature via smart contract.

3.2 Cloud Data Owner Registration

The integrity of the entire system for performing secure communications in cloud computing environment is ensured by only permitting legitimate and verified cloud data users. Every cloud data user has to register itself on the blockchain. The access controlling right i.e. which entities are permitted to perform transactions on the blockchain is with the cloud server 'CS'. When a new cloud data user attempts to register itself in the cloud computing environment, a request 'Req_{reg}' is sent to the cloud server 'CS', that includes the 'NIP_i' and the 'F_i'. The 'F_i' is generated by the equations given below.

$$F_i = f(NIP_i \oplus RND) \quad (1)$$

$$f = NIP_i\%n \quad (2)$$

From the above equations (1) and (2), 'NIP_i' refers to the IP address of cloud data owner node and 'RND' refers to the pseudo random number utilized to hide cloud data owner. When 'CS' receives 'Req_{reg}' it checks in the blockchain of headers. If the cloud data owner is new and no existing cloud data owner block is present for it, a new one is created by the 'CS'.

Followed by which the legitimate data packet flow authority or the cloud server entity identifies the data packet flow from various cloud data owners in a dynamic manner and obtains distinct types and frequencies of data packets, such as, network address of node executing this operation (Node_ip), host name of node executing this operation (Node_name), amount of data starting of API call (Quota_start), amount of data ending of API call (Quota_end), overall storage capacity (Quota_total), and so forth. Furthermore, it identifies the API failure information such as indicates if the API call failed (Failed), includes available failure information of API call (Failure), and so forth from personal cloud dataset. Therefore, the authority tags the collected information about legitimate data packet flows to the legitimate data packet flow authority.

$$\begin{bmatrix} NIP_1F_1 & NIP_1F_2 & \dots & NIP_1F_n \\ NIP_2F_1 & NIP_2F_2 & \dots & NIP_2F_n \\ \dots & \dots & \dots & \dots \\ NIP_mF_1 & NIP_mF_2 & \dots & NIP_mF_n \end{bmatrix} \rightarrow CS \quad (3)$$

From the above formulation (3), the cloud data owner network address executing the operation (Node_ip 'NIP_i') and its corresponding feature values 'F_j' are stored in the cloud server. Next, during the registration process, the legitimate data packet flow classifier or the cloud server classifies the data packets on the blockchain network as it prepares for secured communications in CC environment. In this, the data packet is provided by a labeled legitimate data packet flow analyzer and prepares each blockchain node for training by means of learning algorithm over the transaction network.

$$CS \rightarrow [NIP_iF_i, NIP_iF_{i+1}, \dots., NIP_iF_n\ ] \rightarrow B_i \quad (4)$$

From the above formulation (4), the cloud server stores the legitimate data packet '$NIP_iF_i, NIP_iF_{i+1}, \dots., NIP_iF_n$' on the blockchain network 'B_i'.

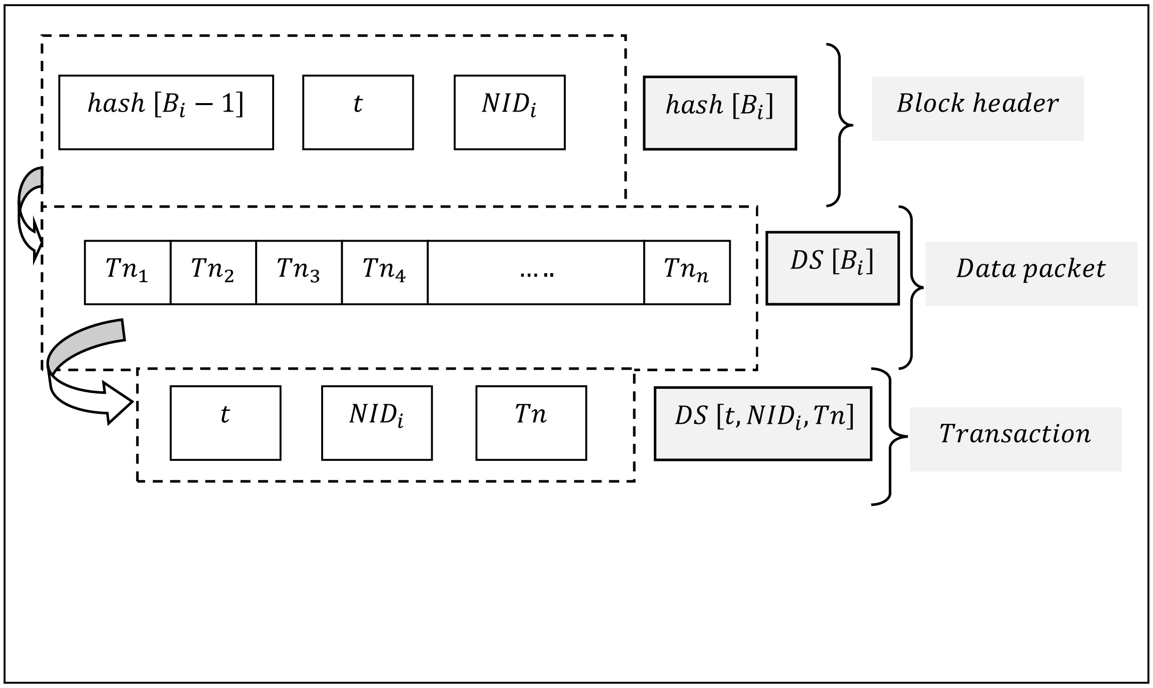
3.3 Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation

Upon successful registration of the Cloud Data Owner made by the cloud server, the registered cloud data owner stores the transactions as given below in the corresponding block for that cloud data owner only.

$$F_i(Tn) \rightarrow F_i(Tn_1 + Tn_2 + \cdots + Tn_n) \quad (5)$$

$$BF_i \rightarrow B(F_n), where\ n \in F_i \quad (6)$$

As given above (5) and (6), each block 'B' consists of one transaction and digital signature is employed that are found to be highly susceptible from malicious users. The transaction block structure is illustrated below in figure 2.



As shown in the above figure, the block structure, the data packet comprises of list of transactions '$(Tn_1, Tn_2, \ldots, Tn_n)$' that records the data packet transaction information. Moreover, the block header comprises of hash value of previous block '$hash[B_{i-1}]$', the cloud data owner node identification 'NID_i', timestamp 't' and index of Boneh–Lynn–Shacham digital signatures '$DS[B_i]$' accountable for creating new blocks in the consensus. The structure of transaction block is then mathematically stated as given below.

$$hash\ [B_i] = f(hash[B_{i-1}], t, NID_i, Tn, DS[B_i]) \quad (7)$$

From the above equation (7), 'f' represent the hash function and during consensus, the digital signature '$DS[B_i]$' is signed to verify the validity of both the current and previous block. Prior to the inclusion of transactions into the disjoint data part of block, they are validated for probable forgery. The testing is carried out via the utilization of the cloud data owner's public key 'PB_{key}' and the private key 'PR_{key}' respectively. This is mathematically stated as given below.

$$PR_{key}(Tn_n) \in F_i(Tn) \tag{8}$$

As when a registered cloud data owner issues a transaction, then it is signed using the 'PR_{key}' as given above (8), and on the other hand if the same transaction is encountered by other entities the 'PB_{key}' is utilized used analogous to 'F_i' for validation of the transactions. Only upon successful validation of the transaction it gets added to the data block and vice versa. The precipitate inclusion in transaction is validated by the proposed method wherein one discrete block for each cloud data owner, whereas in the case of traditional blockchains the blocks carry transactions from diversified entities, therefore improving the latency. As the new block is generated, its header is appended after validation from Schulze Voting Consensus into the blockchain.. The Schulze Voting Consensus is mathematically stated as given below.

$$d[(DO_i),(DO_{i+1})] > d[(DO_{i+1}),(DO_i)] \tag{9}$$

From the above equation (9), in Schulze Voting Consensus, for a pair of cloud data owners and that are associated by at least one data capped (i.e., bandwidth cap or the amount of data packet transferred by a cloud data owner account at a specified throughput over a given time period), the strength of the strongest data capped '$d\left(DO_i, DO_j\right)$' is the maximum strength associating them. If there is no data capped between cloud data owner 'DO_i' and cloud data owner 'DO_j' then, '$d\left(DO_i, DO_j\right) = 0$'. Finally, the block validation is ensured via rule-based smart contract function implemented over a blockchain designed on the basis of a set of rules through which the cloud data owners and cloud users of the smart contract agreed to interact.

$$DPFR = Enc_{PR_{key}}(CDO)[List_n||t] \tag{10}$$

From the above equation (10), the rule-based smart contract function employing the data packet flow record '$DPFR$' is measured based on the cloud data owner private key encryption function '$Enc_{PR_{key}}(CDO)$' and the list of data packet flows '$List_n$' obtained at time 't' respectively. With this rule-based smart contract function are utilized in reporting actions related to data packet requested by the cloud user and permit the cloud data owners to secure and control them, owing to the reason that they will be monitored in a controlled environment. The pseudo code representation of Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation is given below.

```
Input: Dataset '$DS$', cloud data owner '$DO = DO_1, DO_2, \dots, DO_p$', cloud data user '$DU = DU_1, DU_2, \dots, DU_n$', blockchain '$B = B_1, B_2, \dots, B_m$', Data Packet '$DP = DP_1, DP_2, \dots, DP_q$', Cloud Server '$CS$'
Output: Cost efficient and latency improved block validation
Initialize '$m$', '$n$', '$p$', '$q$'
Initialize cloud data owner's public key '$PB_{key}$' and private key '$PR_{key}$'
Begin
    For each Dataset '$DS$' with Cloud Data Owner '$DO$', Blockchain '$B$' and Data Packets '$DP$' to be sent
        // Cloud Data Owner Registration
        For each Cloud Data Owner '$DO$' with Cloud Server '$CS$' and Blockchain '$B$'
            Place a request '$Req_{reg}$' to the cloud server '$CS$' and mathematically stated as given in equations (1) and (2)
            Identification of data packet flow from different cloud data owners by cloud server as given in equations (3) and (4)
            Return registered cloud data owners
        End for
        //Transaction block generation and validation
        For each registered cloud data owners
            Store transactions as given in equations (5) and (6)
            Evaluate structure of transaction block as in equation (7)
            Perform validation using (8) and Schulze Voting Consensus as given in equation (9)
            Formulate rule-based smart contract function as given in equation (10)
            Return generated and validated blocks
        End for
    End for
End
```

Algorithm 1 Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation

As given in the above Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation, the technology of blockchain and cloud computing is integrated to attain improved transaction latency and communication cost. For each cloud data owner to participate in the secure communication process, registration is first performed via cloud server. Second with the registered cloud data owners, a transaction is generated for each data packets and is stored in the block format in blockchain. Third, Transaction Block Generation and Validation are done by means of Boneh–Lynn–Shacham and Schulze Voting Consensus function. With this as only the registered cloud data owners blocks are formulated the transactions latency involved in communication is said to be reduced. Furthermore, with the design of Schulze Voting Consensus-based Block Validation function, the communication cost involved is said to be improved as upon only the satisfaction of Schulze Voting Consensus further process is said to be carried out.

3.4 **Perceptron Neural Network-based secure communication model**

Upon successful validation of blocks communication between cloud data owners and cloud users have to be performed in a secured manner. It means that only the intended cloud users should have the accessibility to view the data and vice versa. The Perceptron Neural Network-based secure communication model being a dual classification machine learning algorithm comprises of a single node or neuron that acquires a row of data (i.e., validated block) as input and predicts a class label (i.e., either proceed with communication or not). This is arrived by evaluating the weighted sum of the inputs and a bias (i.e., initialized to 1) whereas the weighted sum of the input is referred to as the activation. Figure 3 shows the structure of Perceptron Neural Network-based secure communication model.

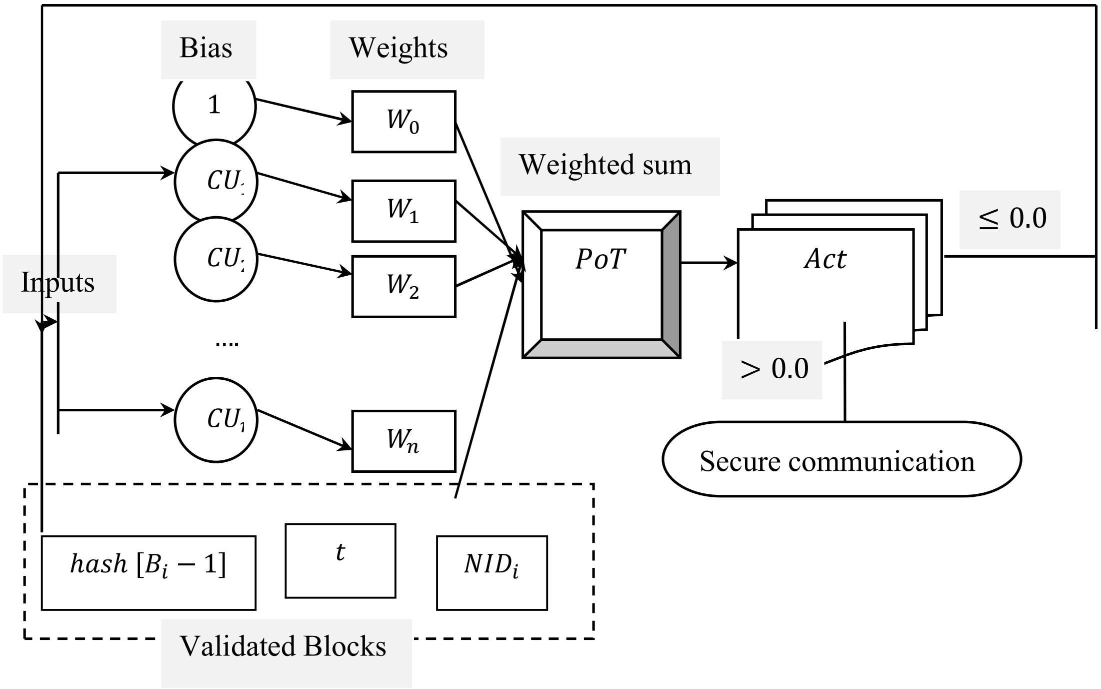


Figure 3 Structure of Perceptron Neural Network-based secure communication model

As shown in the above figure, with the validated blocks and set of cloud users provided as input, initialized bias and arbitrary weights, a secure communication is established by means of Perceptron Neural Network model. Initially, the input comprises of the private key, public key along with the encryption of the data packets for corresponding cloud data owner. Next, point of tangent function is evaluated for initializing session between the cloud data owner and the requested cloud user. Finally, with the activation function, secure communication is said to be established between cloud entities. Let us assume that the cloud data owner 'CDO_i' sends the data packet 'DP_i' with its own private key 'PR_{key}' and 'CDO_i' and public key 'PB_{key}' as received from the cloud server as given below.

$$Private\ Key\ (PR_{key}) \rightarrow\ CDO_i[PR_{key}] \in [1,(n-1)]$$
(11)

$$Public\ Key\ (PB_{key}) \rightarrow\ Private\ Key \rightarrow\ CDO_i[PR_{key}] * G$$
(12)

From the above equations (11) and (12), 'G' represent the integer generator value of prime order. The cloud data owner selects an arbitrary integer '$r \in [1,(n-1)]$' and mathematically formulates the encryption function as given below.

$$Enc = Hash(DP_i)$$
(13)

$$PoT = (CDO_i, CU_i) = K * G$$
(14)

From the above equations (13) and (14), an encryption function ‘Enc’ is first obtained by hashing the data packets of the corresponding cloud data owner ‘$Hash(DP_i)$’ and then the point of tangent ‘PoT’ is formulated to find the communication that has to be established between the cloud data owner and cloud user ‘CDO_i, CU_i’ respectively.

$$\alpha = CDO_i \ mod \ n$$
(15)

$$\beta = K^T(y + \alpha CDO_i) \ mod \ n$$
(16)

From the above equations (15) and (16), to ensure secure communication, by employing the rightmost bits ‘y’ of the encrypted function ‘Enc’, therefore designing an input ‘$Inputs$’ pair of ‘(α, β)’. The activation function is formulated as given below.

$$Act = W * Inputs + B$$
(17)

From the above equation (17), the activation function ‘Act’ is formulated based on the input pair ‘$Inputs = (\alpha, \beta)$’, weight ‘W’, the initialized bias ‘ B’ using a hyperplane in the feature subspace as given below.

$$Act = \begin{cases} 1, if \ Act > 0.0 \\ 0, If \ Act \ \leq 0.0 \end{cases}$$
(18)

From the results of the above equation (18), only upon successful validation results the cloud user is said to be authenticated and valid, therefore ensuring communication or else the cloud user is not authenticated and proceeds with other users. The pseudo code representation of Perceptron Neural Network-based secure communication model is given below.

```
**Input**: Dataset ‘$DS$’, cloud data owner ‘$DO = DO_1, DO_2, \dots, DO_p$’, cloud data user ‘$DU = DU_1, DU_2, \dots, DU_n$’, blockchain ‘$B = B_1, B_2, \dots, B_m$’, Data Packet ‘$DP = DP_1, DP_2, \dots, DP_q$’, Cloud Server ‘$CS$’
**Output**: Robust and secure data communication
**Initialize** integer generator value of prime order ‘$G$’, arbitrary integer ‘$r$’, bias ‘$B = 1$’
**Begin**
**For** each Dataset ‘$DS$’ with Cloud Data Owner ‘$DO$’, Blockchain ‘$B$’ and Data Packets ‘$DP$’ to be sent
```

```
    Generate private key and public key as given in equations (11) and (12)
    For encryption function as given in equation (13)
        Evaluate point of tangent as given in equation (14)
        Obtain validation functions for communicating as given in equation (15) and (16)
    End for
    If '$(\alpha, \beta) \in [1, (n-1)]$ $and\ Act > 0.0$'
    Then validation proved
        Approves key
        Proceed with secure communication
    Else If '$(\alpha, \beta) \in [1, (n-1)]$ $and\ Act\ \leq 0.0$'
        Validation not proved and does not approves with key
        Communication not provided and proceeds with other cloud user
    End if
End for
End
```

Algorithm 2 Perceptron Neural Network-based secure communication

As given in the above Perceptron Neural Network algorithm, with the objective of ensuring robust and secure communication, to start with the private key and public keys are formulated. Followed by which, the encryption function and the point of tangent (i.e., the cloud entities) that needs to be communicated are formulated. With the aid of this point of tangent a row of data (i.e., single neuron) is obtained as input. Finally, validation function is formulated by utilizing perceptrons of hyperplane in the feature subspace. The Perceptron Neural Network algorithm classifies the cloud users into two feature sets, 0 or 1, therefore ensuring data confidentiality.

4. Experimental section

To reveal the efficiency of the proposed Blockchain-based Boneh Lynn Schulze Voting and Perceptron Neural (BBLSV-PN) secured data communication in cloud computing environment, the performance of the system is examined and the simulated results are compared with other existing methods, Blockchain and Smart Contrast-based Data Provenance (BSCDP) [1] and Homomorphic Block Ring Security System (HBRSS) [2] using CloudSim network simulator. This section provides details on the simulation settings and the performance comparison of the proposed method was performed in several scenarios in terms of transaction latency, communication cost and data confidentiality rate with dataset obtained from Personal Cloud Datasets: NEC Personal Cloud Trace (http://cloudspaces.eu/results/datasets).

4.1 Simulation analysis of transaction latency

Latency in a blockchain can be measured due to the time consumed for a particular platform to respond to each transaction. To be more specific it is referred to as the processing time for each transaction ‘’ that is said to be initially broadcasted under the assumption of different transaction number, block size and transaction sending rate. The transaction latency is measured as given below.

$$TL = t_{BInc} - t_{TB} \quad (19)$$

From the above equation (19), the transaction latency 'TL' is measured based on the time the generation of blocks were included 't_{BInc}' and the actual time when the transaction was broadcasted 't_{TB}'. It is measured in terms of seconds (s). The simulation results of all three methods of BBLSV-PN, BSCDP [1] and HBRSS [2] under transaction latency criteria are shown in Table 1 and Figure 4.

Table 1 Tabulation of transaction latency using BBLSV-PN, BSCDP [1] and HBRSS [2]

Number of cloud user data	Transaction latency (s)		
	BBLSV-PN	BSCDP	HBRSS
25	30	40	50
50	35.35	45.35	58.55
75	42.45	52.15	65.35
100	48.35	58.35	68.25
125	51.25	63.25	75.35
150	55.15	68.15	80
175	63.25	73.45	85.25
200	68.55	85.25	91.45
225	72.15	88.15	98.35
250	80	90.35	110.25

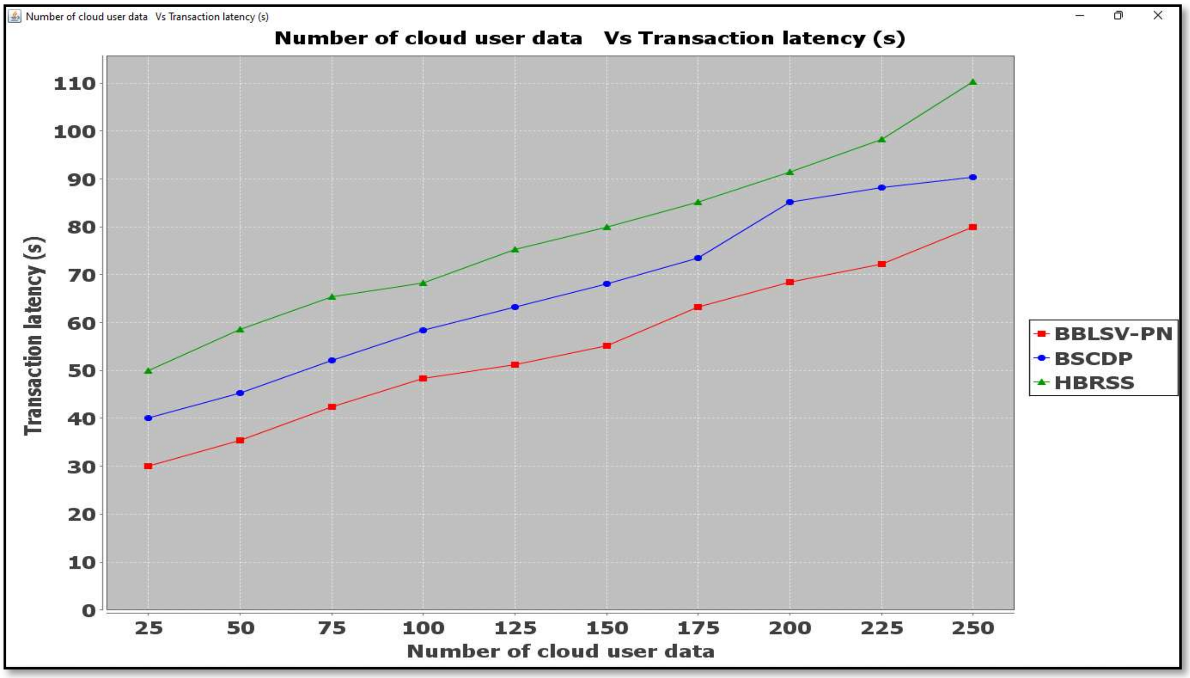


Figure 4 Graphical representation of transaction latency

Figure 4 shows the average transaction latency for creating blocks. Transaction latency on the BBLSV-PN is the lowest, followed by BSCDP [1] and HBRSS [2]. By increasing the value of the number of cloud user data from 25 to 250, the transaction latency criterion of all three methods has also increased. The BBLSV-PN method is significantly better than the BSCDP [1] and HBRSS [2] methods. The reason for the low transaction latency by the proposed CBcA method is that the proposed method uses the following two steps to secure communications and thus reduce transaction latency. It has been used to secure the data packets between the cloud entities in the blockchain platform, and in the second stage, the blocks have been sent using the hashing operation by means of Boneh–Lynn–Shacham digital signatures for creating new blocks in the consensus. The result is that malicious cloud entities nodes are stopped to prevent them from creating additional transactions in the cloud computing environments so that the cloud entities do not perform additional operations to perform transaction and hence consume less transaction latency. Therefore, as can be seen from Table 1 and Figure 4, the proposed method is approximately 18% and 31% better to the BSCDP [1] and HBRSS [2] methods in the transaction latency criterion.

4.2 Simulation analysis of communication cost

While ensuring secure communication in cloud computing environment, a significant amount of cost involved during communication between cloud entities is said to take place. The communication cost in our work is measured on the basis of certain factors involved during secure communication in cloud computing environment. The communication cost is measured as given below.

$$CC = C[NID_i + f + List_n + t] \quad (20)$$

From the above equation (20), the communication cost 'CC' is measured based on the node identity 'NID_i', generation of hash function 'f', list of data packet flows '$List_n$' a timestamp 't' respectively. Communication cost is calculated according to equation (20). The proposed method compared the average communication cost spent on establishing communication between blockchain-based cloud entities at distinct time stamps, the results of which are shown in Figure 5. According to the results obtained from the diagram displayed in Figure 5 and Table 2, it is clear that the communication cost increases linearly with an increase in the number of cloud data users. For example, the communication cost for '$CU = 25$' is smaller than other periods, suggesting the smaller amount of communication cost is ensured as compared with other methods, [1] and [2]. This shows that our proposed BBLSV-PN method has a higher scalability than BSCDP [1] and HBRSS [2] methods in different conditions.

Table 2 Tabulation of communication cost using BBLSV-PN, BSCDP [1] and HBRSS [2]

Number of cloud user data	Communication cost (bits)		
	BBLSV-PN	BSCDP	HBRSS
25	576	783	855
50	675	900	985
75	715	945	1055

100	835	1045	1135
125	915	1090	1185
150	945	1135	1235
175	985	1155	1285
200	1025	1185	1345
225	1085	1245	1365
250	1135	1285	1390

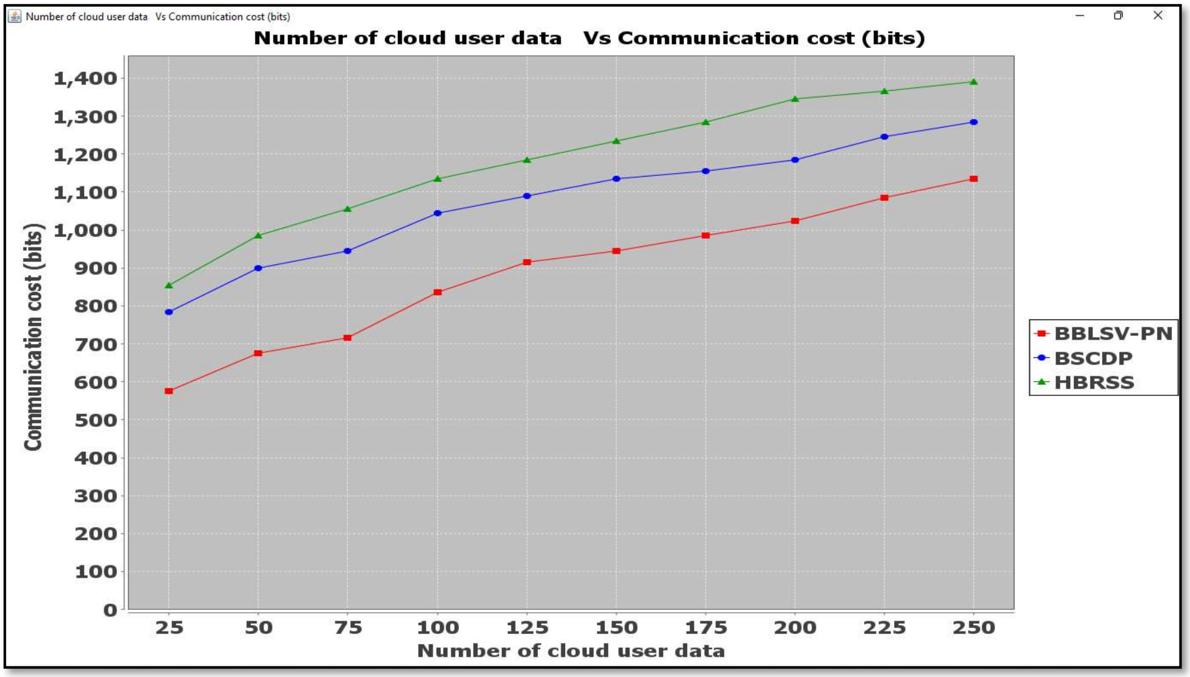


Figure 5Graphical representation of communication cost

In figure 5, for all three methods, the number of requests of data packets is increased by increasing the cloud user data from 25 to 250. In the BBLSV-PN method, when the number of cloud user data is 25, the cost involved in node identity 'NID_i', generation of hash function 'f', list of data packet flows '$List_n$' and timestamp 't' using the BBLSV-PN method was found to be '$128bits, 160bits, 32bits, 256bits$', '$155bits, 195bits, 48bits, 385bits$' using [1] and '$170bits, 215bits, 55bits, 415bits$' using [2] respectively and this shows the superiority of the proposed method. The reason for the superiority of the proposed method over the present two methods is that in the first step, Transaction Block Generation were done using Boneh–Lynn–Shacham and in the second step Schulze Voting Consensus function was utilized for Block Validation separately. This in turn reduced the communication cost using BBLSV-PN method by 18% compared to [1] and 26% compared to [2] respectively.

4.3 Simulation analysis of data confidentiality rate

Secure data communication can be validated by measuring the data confidentiality rate. It is measured based on the cloud data as retrieved by the authorized cloud users with respect to the total number of cloud users' data. Data confidentiality rate is mathematically expressed as given below.

$$DCR = (U_D[Auth])/U_D * 100$$
(21)

From the above equation (21), the data confidentiality rate 'DCR' is measured based on the number of cloud owner data accessed by authorized cloud user '$U_D[Auth]$)' and the number of cloud owner data 'U_D'. It is measured in terms of percentage (%). Finally, the simulation results of data confidentiality are shown in Table 3 and Figure 6. As it turns out, the proposed BBLSV-PN method works better than the other two methods, namely, BSCDP [1] and HBRSS [2]. The reason for the superiority of the proposed BBLSV-PN method over BSCDP [1] and HBRSS [2] methods depends on two reasons, in the first step, the cloud server registers all cloud entities in the blockchain-based environment so that cloud entities that do not have a registration number do not participate in data communication operations, and in the second step, it evaluates individual blocks and authenticates them all so that secure communication is established between them (i.e., between cloud data owner and cloud user).

Table 3 Tabulation of data confidentiality rate using BBLSV-PN, BSCDP [1] and HBRSS [2]

Number of cloud user data	Data confidentiality rate (%)		
	BBLSV-PN	BSCDP	HBRSS
25	92	88	84
50	91.35	87.15	83.25
75	91	86.35	81
100	89.15	85	78.15
125	88.35	84.15	75
150	88	83	73.25
175	86.25	82.15	71
200	85	78	70
225	83.15	75.35	68.55
250	81	73	68

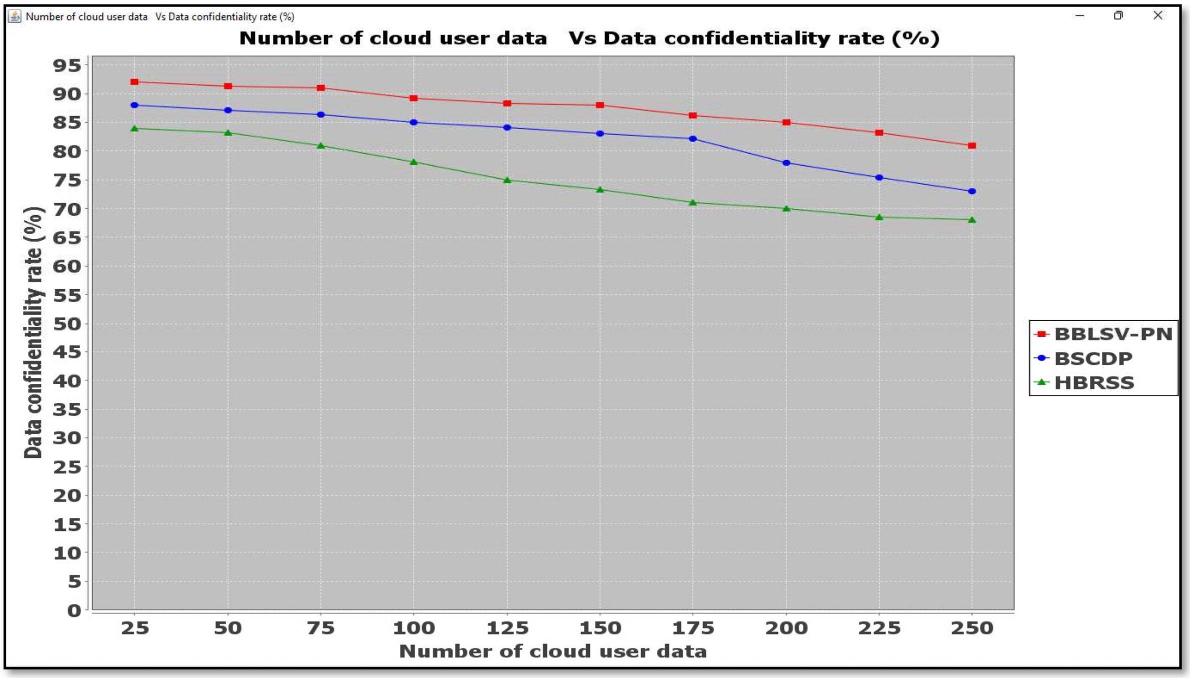


Figure 6 Graphical representation of data confidentiality rate

Figure 6 given above shows the data confidentiality rate measure in terms of percentage (%) over distinct numbers of cloud user data. Also from the above figure it is inferred than increasing the number of cloud user data results in a small decrease in the data confidentiality rate. However, the BBLSV-PN method was found to be better than [1] and [2]. The reason behind the improvement was owing to the application of Perceptron Neural Network algorithm. By applying this algorithm, in the first step distinct private keys were formulated by the cloud server. In the second step, encryption and the point of tangent were formulated separately for each request. Finally, in the third step validation was done by employing perceptrons of hyperplane in the feature subspace, therefore improving data confidentiality using BBLSV-PN method by 7% and 17% compared to [1] and [2] respectively.

5. Conclusion

In this study, we presented a method for a cloud-based secure data communication using blockchain and machine learning. In the proposed method, the data packets of the cloud owner are collected by the cloud server. The data are then transmitted to the requested cloud user only after undergoing two distinct processes, namely, cloud data owner registration and mutual authentication by means of Boneh–Lynn–Shacham-based Transaction Block Generation and Schulze Voting Consensus-based Block Validation. Moreover, when a cloud user requests the data packets from the cloud data owner, the cloud server authenticates the user and shares the requested data packet. Once validation is successful, actual communication is established between the cloud entities by utilizing Perceptron Neural Network-based secure communication model. Furthermore, we demonstrated the robustness of our method and proved that the proposed method is more efficient than the state-of-the-art methods in terms of transaction latency, communication cost and data confidentiality rate using CloudSim simulation.

References

[1] Amrita Jyoti, R. K. Chauhan, “A blockchain and smart contract-based data provenance collection and storing in cloud environment”, Wireless Networks, Springer, Mar 2022 [Blockchain and Smart Contrast-based Data Provenance (BSCDP)]

[2] Hui Xie, Zhengyuan Zhang, Qi Zhang, Shengjun Wei, Changzhen Hu, “HBRSS: Providing high-secure data communication and manipulation in insecure cloud environments”, Computer Communications, Elsevier, Mar 2021 [Homomorphic Block Ring Security System (HBRSS)]

[3] Gagangeet Singh Aujla, Amritpal Singhy, Maninderpal Singhz, Sumit Sharmax, Neeraj Kumar, and Kim-Kwang Raymond Choo, “BloCkEd: Blockchain-based Secure Data Processing Framework in Edge Envisioned V2X Environment”, IEEE Transactions on Vehicular Technology, Oct 2019

[4] Debabrata Samanta, Ahmed H. Alahmadi, Karthikeyan M. P. Mohammad Zubair Khan, Amit Banerjee, Gowtam Kumar Dalapati, Seeram Ramakrishna, “Cipher Block Chaining Support Vector Machine for Secured Decentralized Cloud Enabled Intelligent IoT Architecture”, IEEE ACC ess, Jul 2021

[5] Dai Meiling, Xu Siya, Shao Sujie, Guo Shaoyong, Qiu Xuesong and Xiong Ao, “Blockchain-Based Reliable Fog-Cloud Service Solution for IIoT”, Chinese Journal of Electronics, March 2021

[6] Zia Ullah, Basit Raza, Habib Shah, Shahzad Khan, Han, Abdul Waheed, “Towards Blockchain-Based Secure Storage and Trusted Data Sharing Scheme for IoT Environment”, IEEE Access, Mar 2022

[7] Abdul Rehman, LIU Jian, Muhammad Qasim Yasin and LI Keqiu, “Securing Cloud Storage by Remote Data Integrity Check with Secured Key Generation”, Chinese Journal of Electronics, May 2021

[8] Vaishnavi Moorthy, Revathi Venkataraman, T. Rama Rao, “Security and privacy attacks during data communication in Software Defined Mobile Clouds”, Computer Communications, Elsevier, Feb 2020

[9] Youcef Ould-Yahi, Samia Bouzefrane, Hanifa Bouchene, Soumya Banerjee, “A data-owner centric privacy model with blockchain and adapted attribute-based encryption for internet-of-things and cloud environment”, International Journal of Information and Computer Security, Inderscience, May 2022

[10] Dilip Venkata Kumar Vengala, D. Kavitha, A. P. Siva Kumar, “Secure data transmission on a distributed cloud server with the help of HMCA and data encryption using optimized CP-ABE-ECC ”, Cluster Computing, Springer, Apr 2020

[11] S. Sridhar, S Smys, “Hybrid RSAECC Based Secure Communication in Mobile Cloud Environment”, Wireless Personal Communications, Springer, Nov 2019

[12] Balasubramanian Prabhu Kavin, Sannasi Ganapathy, U. Kanimozhi, Arputharaj Kannan, “An Enhanced Security Framework for Secured Data Storage and Communications in Cloud Using ECC , Access Control and LDSA”, Wireless Personal Communications, Springer, Jun 2020

[13] Ishu Gupta, Ashutosh Kumar Singh, Chung-Nan Lee, Rajkumar Buyya, “Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic

Review, Analysis, and Future Directions", IEEE Access, Jul 2022

[14] Wenjuan Li, Jiyi Wu, Jian Cao, Nan Chen, Qifei Zhang and Rajkumar Buyya, "Blockchain-based trust management in Cloud Computing systems: a taxonomy, review and future directions", Journal of Cloud Computing: Advances, Systems and Applications, Springer, Oct 2021

[15] Pan Yang, Naixue Xiong, Jingli Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey", IEEE Access, Jul 2020

[16] Ch. V. N. U. Bharathi Murthy, M. Lawanya Shri, Seifedine Kadry, Sangsoon Lim, "Blockchain Based CC: Architecture and Research Challenges", IEEE Access, Nov 2020

[17] Mohamed Amine Ferrag, Leandros Maglaras," DeepCoin: A Novel Deep learning and Blockchain-based Energy Exchange Framework for Smart Grids", IEEE Transactions on Engineering Management, November 2020

[18] Maryam Ataei Nezhad, Hamid Barati, Ali Barati, "An Authentication-Based Secure Data Aggregation Method in Internet of Things", Journal of Grid Computing, Springer, Jul 2022

[19] S. Immaculate Shyla, S. S. Sujatha, "Efficient secure data retrieval on cloud using multi-stage authentication and optimized blowfish algorithm", Journal of Ambient Intelligence and Humanized Computing, Springer, Jan 2021

[20] Yushu Zhang, Ping Wang, Liming Fang, Xing He, Hao Han, and Bing Chen, "Secure Transmission of Compressed Sampling Data Using Edge Clouds", IEEE Transactions on Industrial Informatics, Jul 2019