Volume: 5 Issue: VI Month of publication: June 2017

DOI:

Volume 5 Issue VI, June 2017
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Improving Existing Data Security Standards in Cloud Computing Using Trust Based Machine Learning

Aditya Chellam[1]

[1]*School of Computer Science and Engineering, VIT University, Vellore, India*

***Abstract:** The onset of the Big Data phenomenon, has brought to light the need to store and process data externally for effective and efficient computation. Cloud computing is a technology that has enabled individual users and organizations alike to implement such a functionality. Currently, a large percentage of the data being generated is stored on clouds and the number of organizations opting for cloud based technologies is constantly on the rise. With such growing numbers accessing and utilizing cloud resources, data security has become a major cause of concern. Traditional methods of cloud computing are becoming obsolete and ineffective with each technological breakthrough and data is thus highly subjected to getting corrupted or hacked. Machine Learning algorithms can be implemented to program the security mechanism such that the cloud is able to verify and secure the data with greater efficiency and improve the security predictions as more and more data gets accumulated. An AI (Artificial Intelligence) driven framework for cloud computing, can not only handle the current data traffic but is also a viable framework for the future as it learns and improves itself constantly as the accreted knowledge base increases. This paper implements Random Forest machine learning algorithm to improve the data security in cloud computing.*

***Keywords:** Cloud Computing, Big Data, Artificial Intelligence, Random Forest Tree, Machine Learning, AES*

I. INTRODUCTION

Cloud computing is the practice of making use of external resources hosted on the internet for computational purposes. The “cloud”, is essentially a data center or service provider station which provides various resources such as Software (SaaS – Software as a Service), Infrastructure (IaaS – Infrastructure as a Service), and Platform (PaaS – Platform as a Service) and makes these resources accessible to the general populace by hosting them over the internet. The users can then make use of the resources based on the guidelines as specified by the cloud provider. Data security in the cloud computing context involves the implementation of protective privacy measures in order to safeguard the data stored in the cloud from being accessed by unauthorized sources. The data stored in the cloud is not only at risk of being accessed by unverified sources but is also vulnerable to data corruption. The data security domain is inclusive of all such risks to the data stored in the cloud. The data being generate is on an exponential rise, leading to large amounts of both structured and unstructured data which can loosely be termed as big data. Observing the trends in big data leads us to a logical and intuitive conclusion that, the best way to store such huge multitudes of data would be to store it in such a way that it is self-monitored. Doing so would not only easy the data verification and processing loads but would also improve the overall speed and thereby the efficiency of cloud computing.

Machine Learning (ML) a sub-domain of AI (Artificial Intelligence) enables computers to learn without having to be being explicitly programmed. ML primarily aims at developing programs which can make correct and optimum predictions when exposed to new data based on past experiences to similar data. ML is quite similar to data mining, however the key difference between the two is that, unlike data mining, where human comprehension is utilized to make decisions when encountered with new data, machine learning makes use of patterns present in existing database to adjust itself in such a way so as to make an informed decision which would help incorporate the new information into the existing framework. Machine learning algorithms are loosely classified into two sub- domains based on the learning technique used. The sub-domains are, supervised learning and unsupervised learning. In unsupervised learning algorithms, the machine learns by drawing inferences from existent datasets. In the case of supervised algorithms can apply what has been learned in the past to new data. Although quite similar to each other, both these learning techniques are fundamentally different from each other.

II. LITERATURE REVIEW

Various existing forms of data security measures have been surveyed for this paper. Firstly, the various problems that are currently existing in cloud computing technologies, such as data validation, data encryption, data confidentiality, integrity of data and data availability have been identified, discussed and analyzed. Various Data Security measures that uses traditional paradigms have been discussed by various authors [7] [11] [12]. Apart from these, various storage security techniques have also been discussed. The findings from each paper survey have been tabulated in the following table.

A. Survey Table

TABLE I
TABULAR REPRESENTATION OF SURVEYED DATA

Sr. No.	Author Name	Domain Addressed	Description	Algorithm Used	Advantage
1	Mostapha Derfouf	Storage security	Data stored in cloud has following security issues: Data Loss / leakage, Service / Account hijacking, DDOS (Distributed denial of service) attack	-	-
2	Komal Singh Gill and Anju Sharma	Green Cloud Computing	Security appliances used in the form of physical machines or by virtualization. IDPS is used to reduce the overall power consumption of the virtual network	Intrusion Detection and Prevention System (IDPS)	Amount of power consumed is reduced
3	Shohreh Hosseinzadeh et al.	Privacy in cloud computing	Classification is done based on how the diversification/obfuscation techniques are used to enhance the security in cloud computing environment.	Obfuscation and Diversification	Impedes malware from causing harm in different domains
4	Pin Zhang et al.	Access Control to Cloud	Based on role-based access control, a Hierarchical role is proposed. For the dynamics of Cloud computing environment, the trusted level of the users is divided and updated in real time.	Trust Role-Based Access Control (RBAC),	Data Protection from malicious attackers
5	Kamal Kumar Chauhan et al.	Data Security	Homomorphic Encryption method is used to encrypt and secure the data	Homomorphic Encryption	Data security improved
6	Stuti Srivatsava et al.	Security in cloud computing	Various cloud computing paradigms, standards and guidelines discussed	-	-
7	Zoltán Balogh et al.	Cloud Computing Model	Various cloud computing models as defined by The Organization of National Institute of Standards and Technology (NIST) discussed in detail	-	-
8	K.B.Priya Iyer et al.	Data Security	A detailed analysis of data privacy, confidentiality and integrity	-	-
9	Ahmed Albugmi et al.	Data Security	Provides an insight on data security aspects for Data-in-Transit and Data-at-Rest for all levels of SaaS, IaaS and PaaS	-	-
10	Bin Feng et al.	Storage Security	Bidirectional Verification Technique implemented to verify the data to be stored	KeyGen, SigTanGen, Register algorithms	Credibility of Data stored is increased

11	Oinam David Singh et al.	Network Security	A nucleic filter applied between data center and end user to improve security	Nucleic Filter	Helps detect or destroy illegitimate users or attackers
12	Shankar Nayak Bhukya et al.	Data Security	Provable data possession (PDP) model that allows a client having stored data at an untrusted server to verify that the server possesses the original data without retrieving it proposed	Provable Data Possession (PDP model)	Data security increased
13	Rohan Raj Gupta et al.	Storage Security	Proposed model encrypts user data when it is uploaded to the servers, and reduces the cost overhead of encryption as we are not using standalone hardware servers for encryption which waste resources even if they are not in use	Container Clustering	Cost overhead of encryption reduced
14	Ashalatha R et al.	Cloud Network Virtualization	Virtualization of resources associated to the cloud network proposed	Network Virtualization Technology and other encryption algorithms	Maximum resource utilization and high flexibility
15	I. Ennajjar et al.	Data Security	Data security issues of availability, breach and loss of data discussed	-	-
16	Neha A Puri et al.	Data Security	The deployment of Application on the Cloud and increased security level by implementation ECC Algorithm, Digital Signature and Encryption explored	ECC (Elliptic Curve Cryptography)	Low CPU utilization and reduced time for encryption
17	S Raju et al.	Data Security	Data security provided by implemented Cramer-Shoup Cryptosystem	Cramer – Shoup Cryptosystem	Data is secured
18	Sushil Kr Saroj et al.	Data Security	A scheme that uses threshold cryptography in which data owner divides users in groups and gives single key to each user group for decryption of data and, each user in the group shares parts of the key proposed	Threshold Cryptography	Data is secured

III.PROBLEM STATEMENT

Improving the efficiency of existing data security techniques while taking into account the increasing amount of data being generated is the need of the hour. Data validation is an important attribute when deciding the cloud architecture by IaaS (Infrastructure as a Service) providers. At the highest level of abstraction, there exist three perquisites for secure cloud communication: data confidentiality, data integrity and data availability [1][2]. The principal focus of the proposed model in the subsequent section is improving data availability without compromising the effectiveness of the other two sub-domains. Multiple nodes accessing cloud resources have an inherent encryption-decryption overhead necessary for data security. The proposed model aims at reducing these incurred overheads to improve data availability. Another possible constraint that may arise with multiple nodes in a closely coupled cloud network, simultaneously requesting access to cloud resources is deadlocks, however dealing with deadlocks is beyond the scope of this paper.

IV. PROPOSED METHOD: TRUST BASED VALIDATION TECHNIQUE

The proposed model is a trust based validation method for data security that reduces data validation time based on past interactions with the client node in concern. Each node is initially assigned a trust factor which acts as an index of validation for node in concern. With each data exchange the trust factor of the node is updated in real time. If the data provided by the node is found to maintain data integrity and no abnormalities are detected after validation, the trust factor is incremented progressively over a present number of attempts. Correspondingly any discrepancies in the received data or any unsuccessfully handled request has a negative impact on the trust factor of the node. The assigned trust factors show an aggregated measure of the performance and validity of each node.

Using Random Forest machine learning algorithm, a virtual “memory” of the history of data validation results of each node is maintained. This memory provides the required advice parameters necessary for deciding the number of rounds of encryptions necessary. The nodes performing consistent legitimate data transfers, achieve a reasonably good trust factor over time. A better trust factor implies that the data received from the node in concern, is mostly consistent. This estimation is quantified mathematically using random forest algorithm. Thus, a good trust factor ensues fewer rounds of encryption saving data validation overheads [19]. Fewer rounds of encryption at the client node sides, results in fewer number of decryption at the cloud server side.

A. *Random Forest Algorithm*

The node trust factors are updated based on the result of decision tree structures for each node. Multiple weak decision trees function in parallel as single unit, by taking an aggregated of all the trees based on majority voting. This is the fundamental principle behind random forest learning algorithm. To avoid contextual biases while voting, each of the decision trees is fed with a slightly varying input parameter. In order to get a robust estimate of the trust factor Bootstrap Aggregation (Bagging) is used for regression [20]. For each tree in the forest, a bootstrap sample from Sample Space ‘S’ is selected, where $S(i)$ denotes the i^{th} bootstrap. At each node of the tree, some subset of the features $f \subseteq F$ are randomly selected; where F is the set of features; integrity, confidentiality, consistency etc. The node then splits on the best feature in subset f rather than F. The subset f is very small as compared to F and thus reduces the number of parses. The feature to be split is trust factor and is the most computationally expensive task in the decision-making process.

1) *The pseudocode for Random Forest Machine Learning is illustrated as follows*

a) Algorithm to determine Rounds of Encryption

b) Precondition: Training set S := $(x_1, y_1),(x_2,y_2,), . . . ,(x_n, y_n)$, features F, and number of trees in forest B.

```
function RandomForestTrustDetermination(S , F)
H ← Ø
for i ∈ 1, . . . , B do
S (i) ← Bootstrap sample from S
hi ← RandomTreeLearning(S (i) , F)
H ← H
∪ {hi}
end for return H
end function
function RandomTreeLearning (S , F)
At each node:
f ← very small subset of F
Split on feature {trust-factor} in f
return Learned Tree
end function
```

B. *Encryption - Advanced Encryption Standard (AES)*

The data to be transmitted has to be encrypted at client side before transmission to the cloud server for safeguarding the data packets. In accordance to our proposed model AES is a suitable standard as it is an iterative encryption model and can have varying rounds of encryption depending on the size of the key (10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys, etc.) [17][10]. Each round of the AES is divided into four sub-processes – Byte Substitution, Shift Rows, Mix Columns, and Add Round Key. AES uses. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

1) *The pseudocode for the client side encryption is as follows –*

a) Algorithm for AES client-side Encryption

```
Input:  Table T and key k
Output: table T modified
function AES_encrypt (T, k)
Start
    KeyExpand (k, Tk);
    AddRoundKey (T, Tk[0]);
    for (i = 1; i <$n_r$; i + +)
        Round (T, Tk [i]);
    FInalRound (T, Tk [nr]);
End
End function//Definition of Round Function
Input: table T and a tower key Z
Output: Table T modified
function Round (T, Z)
Start
    SubBytes (T);
    ShiftRows (T);
    MixColumns (T);
    AddRoundKey (T, Z);
End
End function
```

xxii) The key length and therefore the corresponding number of rounds of encryption, is directly determined based on the trust-factor advice parameter obtained as a result of the aforementioned random forest algorithm.

C. Server Side Decryption

The received data in the cloud is decrypted using the same key used for encryption. The encryption routine is reversed and reordered to produce a decryption algorithm using InverseSubBytes transformations, InverseShiftRows, InverseMixColumns and AddRoundKey.

1) *The pseudocode for the server side decryption is as follows –*

a) Algorithm for AES server-side Decryption

```
AESdecryption(T, K)
{
    KeyExpand(K, RoundKeys);
    /*First Round*/
    AddRoundKey(State, RoundKeys[Nr]);
    for (r=Nr-1; i>0; r--) {
        InverseShiftRows(T);
        InverseSubBytes(T);
        AddRoundKey(T, RoundKeys[r]);
        InverseMixColumns(T);
}
/* FinalRound */
InverseShiftRows(Out);
InverseSubBytes(Out);
AddRoundKey(Out,RoundKeys[0]);
}
```

The cloud infrastructure required for this experiment was built on a custom simulator using Java Platform. The environmental specifications of the system running this application is a laptop with 4GB RAM running Windows 10 operating system. Weka 3.8 tool has been used to implement the Random Forest (RF) algorithm to train the data for detecting discrepancies in the transmitted data. The data features are represented as vectors which are fed to the tool. A 10 point Likert scale is used to map the trust value of the nodes. 10 being theoretical limit, indicating that if a node has a trust factor of 10 it essentially can be admitted without a single round of validation. Initially a 128-bit key is used for encryption; 10 rounds. A total of 100 sessions of communication were conducted. During randomly selected sessions erroneous data was transmitted intentionally. The trust factor of the corresponding node is observed and it is noted that on erroneous data transmission the trust value immediately falls increasing the number of rounds of encryption by corollary. The trust factors of each node before and after erroneous data transmission are represented in histogram notation in the following section. Floor values of the updated trust factors are taken to obtain discrete results.

From this experiment, it is evident that the rounds of encryption vary inversely as the trust factor decreases. With time, if a node continues to transmit erroneous data to the cloud, the trust factor of the node in concern drops drastically and reinstating trust becomes increasingly difficult.

V. EXPERIMENTAL SETUP

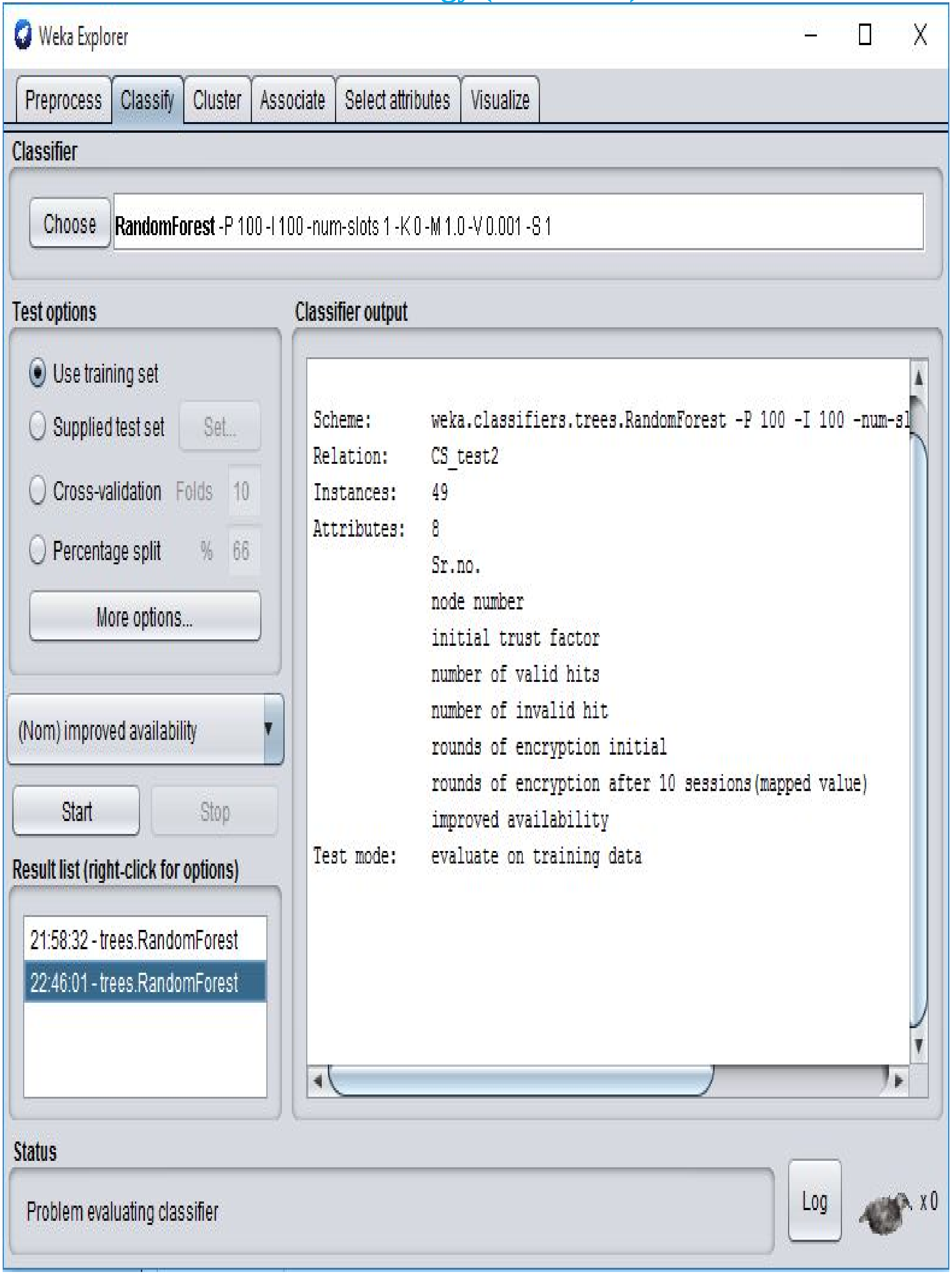
The tool required is Weka 3.8 along with Netbeans IDE 8.2 for CloudSim simulation. The minimum system specifications required are:

TABLE II. MINIMUM SPECIFICATIONS REQUIRED

Processor required	Pentium 520 MHz
RAM required	1024 MB or More
Disk space required	1024 MB Disk Drive
Netbeans IDE version	8.0 or above
Internet Connection	Nil

VI. RESULTS AND DISCUSSIONS

The following histogram representations clearly distinguish the difference in validation overheads with and without the implementation of the proposed model. A snapshot of the model in function after 25 sessions of data transfer is noted to observe the changes in trust factor values and rounds of encryption. From the observed details, it is clear that more trusted nodes require lesser rounds of encryption and decryption and thus time required for data validation is greatly reduced. Improving the time attribute of security in cloud computing helps improve data availability and thus the overall cloud security framework as a whole.



Classifier

RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) improved availability

Result list (right-click for options)

21:58:32 - trees.RandomForest
22:46:01 - trees.RandomForest

Classifier output

```
Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-sl
Relation:     CS_test2
Instances:    49
Attributes:   8
              Sr.no.
              node number
              initial trust factor
              number of valid hits
              number of invalid hit
              rounds of encryption initial
              rounds of encryption after 10 sessions(mapped value)
              improved availability
Test mode:    evaluate on training data
```

Status

Problem evaluating classifier

Fig 1. Weka tool with Random Forest Algorithm selected

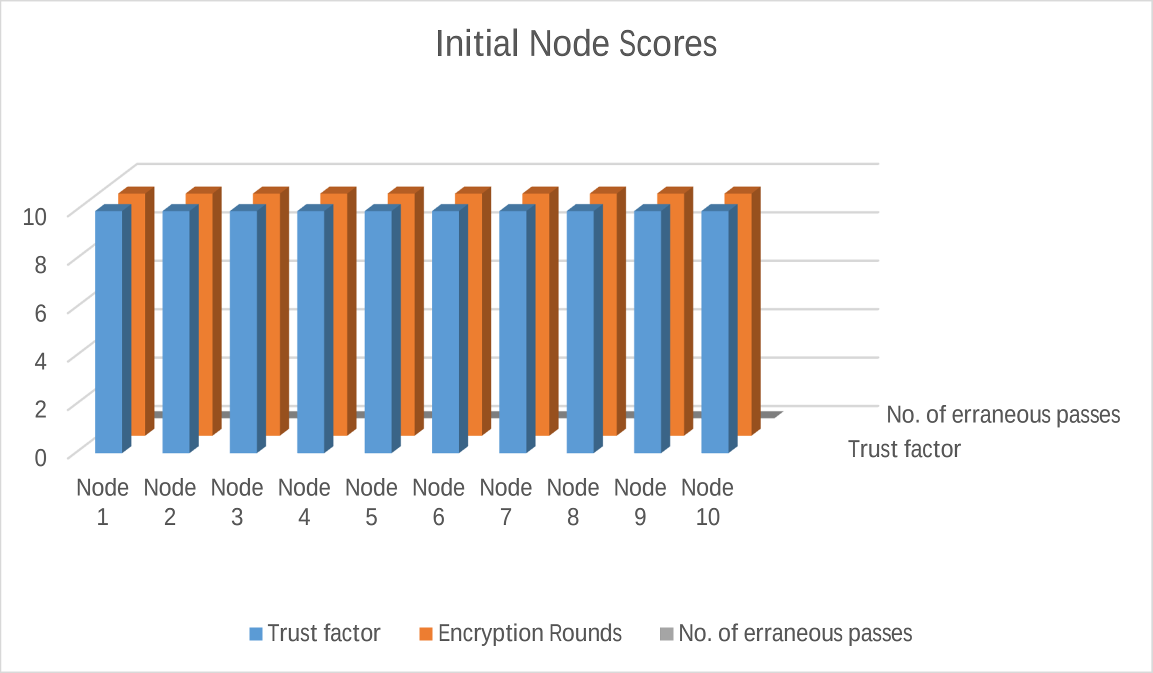


Fig 2. Initial Trust factor values

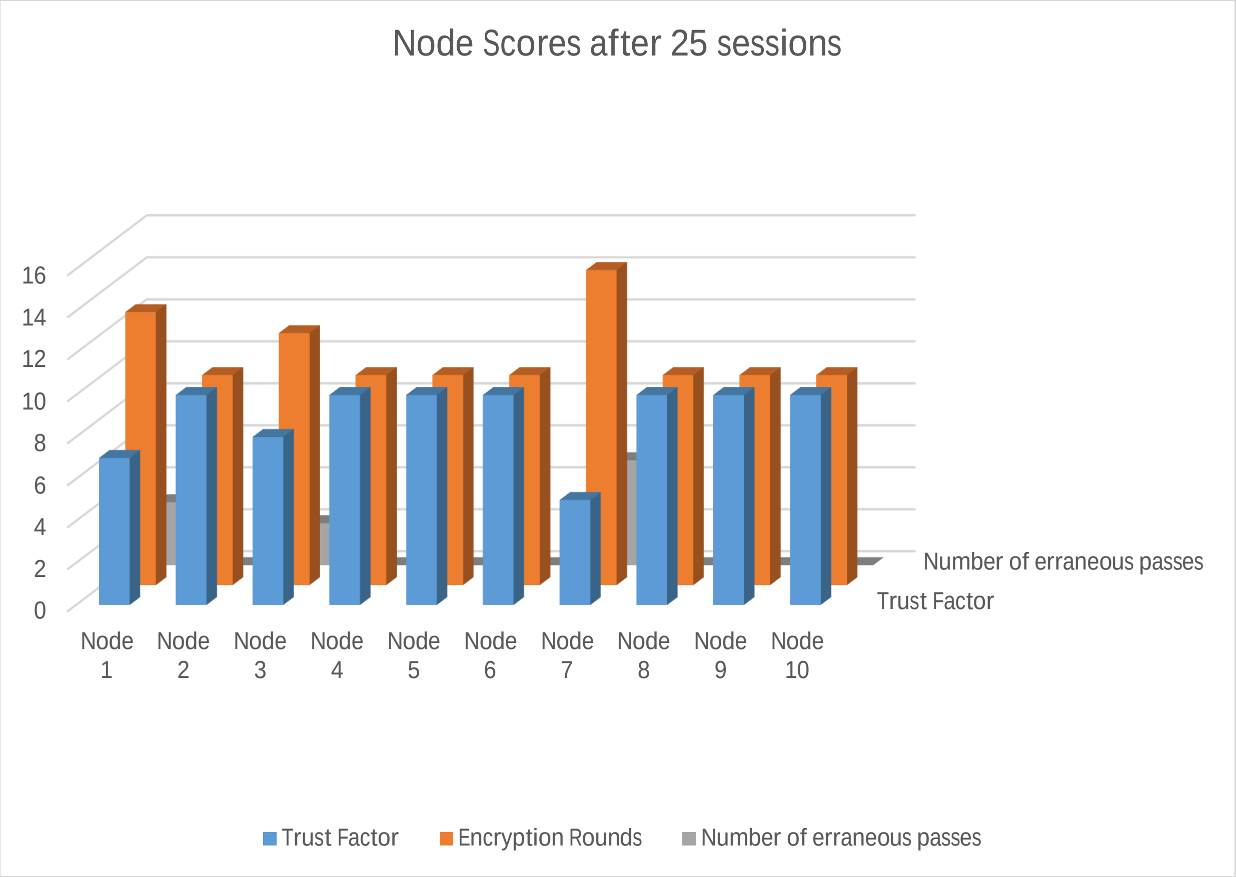


Fig 3. Updated and discretized trust factor values after 25 sessions

VII. CONCLUSION

Summing up, the model presented in this paper, reduces the data validation overheads in cloud communications by assigning trust values to individual nodes which can vary the rounds of encryption, ensuing less validation time for more trusted sources of data. This paper proposes a method to improve data availability to improve cloud based communication without compromising data security. For future work, a mechanism for handling simultaneous requests for cloud resources that may lead to deadlocks in closely coupled networks, can be developed for further improving cloud based communication.

VIII. ACKNOWLEDGMENT

I would like to thank all the people who have motivated and helped me most throughout my project especially my colleagues who, by exchanging their own thoughts and providing valuable input made it possible to complete the paper with all accurate information.

REFERENCES

[1] D. W. K. Tse, “Challenges on Privacy and Reliability in Cloud Computing Security.”, pp. 1181-1187.

[2] P. Zhang, J. Xu, H. Muazu, and W. Mao, “Access Control Research on Data Security in Cloud Computing,” pp. 873–877.

[3] R. Ashalatha, “Network Virtualization System for Security in Cloud Computing,” pp. 346–350, 2017.

[4] M. O. P. V. College and F. Women, “ANALYSIS OF DATA SECURITY IN CLOUD COMPUTING.”, International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16), 2016.

[5] A. Albugmi, R. Walters, and G. Wills, “Data Security in Cloud Computing,” pp. 55–59, 2016.

[6] S. L. Albuquerque and P. R. L. Gondim, “Security in Cloud- Mobile Health,” pp. 37–44, 2016.

[7] Z. Balogh and M. Tur, “Modeling of Data Security in Cloud Computing,” 2016.

[8] S. Bhukya and S. Pabboju, “Data Security in Cloud computing and Outsourced Databases,” pp. 2458–2462, 2016.

[9] R. R. Gupta, “Data Storage Security in Cloud Computing Using Container Clustering,” 2016.

[10] T. I. E. Qiu and S. Member, “An Efficient Protocol With Bidirectional Verification for Storage Security in Cloud Computing,” vol. 4, 2016.

[11] M. K. Sarkar, “A Framework to Ensure Data Storage Security in Cloud Computing,” pp. 3–6, 2016.

[12] L. Wr, Q. Wkh, G. Wkrl, and G. H. F. Jpdlo, “& RPSXWLQJ (QYLURQPHQW,” vol. 6, pp. 3762–3765, 2016.

[13] Srivastava, Stuti, and Prem Sewak Sudhish. "Security in cloud computing systems: A review of challenges and solutions for security in distributed computing environments." Systems Conference (NSC), 2015 39th National. IEEE, 2015.

[14] N. Bohlol, “Systematic Parameters vs . SLAs for Security in Cloud Computing,” 2015.

[15] M. Derfouf, “Vulnerabilities and storage security in Cloud Computing,” 2015.

[16] K. S. Gill and A. Sharma, “IDPS based Framework for Security in Green Cloud Computing and Comprehensive Review on Existing Frameworks and Security Issues,” 2015.

[17] S. Hosseinzadeh, S. Hyrynsalmi, M. Conti, and V. Lepp, “Security and Privacy in Cloud Computing via Obfuscation and Diversification : a Survey,” 2015.

[18] I. International, C. Conference, I. Information, and T. Technology, “International International Conference Conference on Information Information Technology Technology,” pp. 206–209, 2015.

[19] A. M. Khan, S. Ahmad, and M. Haroon, “A Comparative Study of Trends in Security in Cloud Computing,” 2015.

[20] C. Technology, S. K. Saroj, G. Noida, S. K. Chauhan, A. K. Sharma, and S. Vats, “Threshold Cryptography Based Data Security in Cloud Computing,” 2015.

[21] I. It, “Deployment of Application on Cloud and Enhanced Data Security in Cloud Computing using ECC Algorithm,” no. 978, pp. 1667–1671, 2014.

[22] L. Oderudwru et al., “Security in Cloud Computing approaches and solutions,” pp. 57–61, 2014\

[23] Y. M. Sirajudeen, “DATA SECURITY IN CLOUD COMPUTING USING CRAMER – SHOUP,” pp. 343–346, 2014.

[24] Sarvabhatla, Mrudula, M. Giri, and Chandra Sekhar Vorugunti. "A robust ticket-based mutual authentication scheme for data security in cloud computing." Data Science & Engineering (ICDSE), 2014 International Conference on. IEEE, 2014.