DOI 10.12694/scpe.v20i2.1529
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 2, pp. 349–364. http://www.scpe.org © 2019 SCPE

TRUST FACTOR BASED KEY DISTRIBUTION PROTOCOL IN HYBRID CLOUD ENVIRONMENT

VELLIANGIRI S. [*] RAJAGOPAL R. [†] AND KARTHIKEYAN P. [‡]

Abstract. In the Hybrid cloud deployment model, security is essential to restrict access while using resources such as virtual machine, platform, and application. Many protocols were developed to provide security via the cryptography technique, but these protocols rarely considered the trust factor which is an essential factor for cloud security. The existing Elliptic Curve Cryptography and Diffie Hellman key distribution mechanism failed to stress the trust factor, and further, they have provided not only higher complexity but also lower security and reliability. The proposed method comprised two stages: first stage, Group Creation using the trust factor and develop key distribution security protocol. It performs the communication process among the virtual machine communication nodes. Creating several groups based on the cluster and trust factors methods. The second stage, the ECC (Elliptic Curve Cryptography) based distribution security protocol is developed. The proposed Trust Factor Based Key Distribution Protocol reduced error rate, improve the key computation time and maximize resource utilization.

Key words: Cloud Computing, Elliptic Curve Cryptography, Clustering, Trust Factors, Key Distribution

AMS subject classifications. 94A60, 68M14

1. Introduction. Cloud computing has emerged as a pivoted new field, differentiated from conventional computing by its focus on software as services, platform as services and infrastructure as services, these services are implemented by Cloud deployment models such as public cloud, private cloud or hybrid cloud. Cloud computing has added to the advances in computational and correspondence innovations - making financially achievable the combination of various clusters of heterogeneous network resources and services which, thus, have prompted the improvement of extensive large scale distributed system. A group key exchange protocol permits a set of people to agree upon a shared secret session key over an internet network [1, 2].

Cloud computing virtual resource can be characterized into three types such as compute, storage and network. Compute resource permit you to start virtual machines on cloud infrastructure. The main challenges of task scheduling in compute clouds are its highly dynamic environment, where the computing resources have their availability, access policies, security, reliability, etc. [3, 4]. Cloud computing has often jointly supported collaborative computing projects on the internet. Maximum of these projects have stringent security requirements. The capabilities for cloud systems and networks, such as broadband capabilities and distributed intelligence can greatly enrich reliability and efficiency, but they might also create much vulnerability if not deployed with the appropriate security controls. Providing security for such a large system may seem an unfathomable task, and if done incorrectly, can leave utilities open to cyber-attacks [5, 6].

Communicate among the group of members in securely and tried to reduce the complexity of informing the group key. The secure communications in the group or large group are more complex than Peer-to-peer communication because of the adaptability issue of the group key. Specifically, the expense of key establishment and renovating are generally pertinent to the size of the group and consequently turns into a performance bottleneck in attain scalability. To address this issue, our proposed approach named as the trust factor based key distribution protocol that combines the features of ECC key exchange protocol with the trust factor. ECC and Diffie Hellman key exchange protocols do not consider the trust factor in that which is relating to a ”firm belief ” with a treat like trustworthiness, reliability, and ability of the element for multicast communication in distributed computing [7, 8].

The primary goal of a cloud environment is increasing the domain to domain interactions, a secure environment and to ensuring the confidentiality of others. To achieve this ”trust,” notion needs to be considered so that the trustworthiness makes the geographically distributed method more reliable and attractive. The trust

[*] Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad - 501401, Telangana, India (velliangiris@gmail.com).

[†] Department of Computer Science and Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi -600062, Chennai, Tamilnadu, India (rajagopalrmail@gmail.com).

[‡] Department of Computer Science and Engineering, Karunya Institute of technology and Sciences, Coimbatore - 641114, Tamilnadu, India (nrmkarthi@gmail.com)

factor is an essential factor for cloud security. The above issues were resolved by using our proposed method of trust-based cloud security. In our proposed method has considered the priority of trust factor for enhancing the parameters of reliability and security. Hence to reduce this drawback in the existing protocols like ECC and Diffie Hellman key exchange, the trust factor based distribution security protocol using ECC is proposed, these trust factors based on key distribution security protocol, have been utilized to select the group leader among the communication nodes.

After that, the ECC method has been exploited in the secure communication process among group members. When compared to ECC and Diffie Hellman key exchange, the proposed method has given high performance in security. The proposed system provides improved scalability due to the trust factor Key Computation [9, 10].

Main contributions of this research work are summarized below:

- Trust Factor Based key distribution protocol is designed that helps the decision making during the key distribution.
- Trust Factor Based key distribution protocol performances are evaluated based on the theoretical analysis and experimental analysis.
- The objective of the research work is to minimize the error rate and computation time and maximize the resource utilization using Trust Factor Based key distribution protocol

The remainders of the paper are structured as follows, in section 2 crucial secure exchange protocol are reviewed. Section 3 we discussed the proposed system and implementation. In Section 4, we analyze and prove the correctness and safety of the proposed protocol and comparative study also reported in this section. Finally, we conclude the paper in Section 5.

2. Related Works. This section provides various recent related works regarding the secure key exchange protocol are reviewed. Wei et al. proposed a compiler which changes over secure group key exchange protocol into a secure passive keyless entry protocol. This method uses the labeled signature, public key encryption, the hash function. The proposed compiler is the main provably secure compiler without irregular prophets. As far as efficiency, the proposed protocol simply improved the two rounds of communications to the unique group key exchange protocol, which suggests that the derived password-authenticated group key exchange protocol is a static-round protocol as long as the secret group key exchange protocol is a static-round protocol [11]. Gonzales et al. proposed the model of security assessment for Infrastructure as a Service; this plan is confined to the single cloud service model, infrastructure as a service. The layers of the software stack underneath the guest operating system are coming under the control of the Infrastructures as service cloud service providers, key exchange security important in the software as service cloud service model [12].

Li et al. [13] introduced the blockchain based security model for centralized cloud storage. They found that the model has been contrasted with other two existing traditional model in respect of network transmission delay and security. Moreover, the system performance of conventional cloud design has been enhanced by the modified evolutionary algorithm which decreases the expenses on transmitting and replicas scheduling. Additionally, the proposed architecture has lower transmission delay compare with the genetic algorithm. Recently various exchange protocol proposed in a cloud environment concerning the healthcare environment. Challa et al. introduced a novelty of Secure Authentication Scheme (SAS) for healthcare in wireless sensor networks. To figure out the drawbacks in the LiuChung's existing method [14], the proposed framework enables a lawful client to modify/update the biometrics and password without prior communicating the trusted third party authority. Besides, it also permits a revocation scheme for disobeying nodes in the wireless sensor network. The proposed system uses BurrowsAbadiNeedham logic as well as the random oracle model. Additionally, the proposed scheme prevents replay attacks with the help of the widely-used AVISPA tool. The low communication costs and computation and along with high security make the proposed method fit for an extensive range of healthcare projects [15, 16].

Barman et al. designed a basic session-based exchange protocol that was worked like as biometric information of two conveying parties are pooled together to fabricate a cryptographic architecture are called a mutual locker. [17]. Zhou et al. have reported the key exchange challenges of k-nearest neighbors query over the encoded database in cloud services vendor. The revealed plan to accomplish secure outsourcing storage and k-nearest neighbors question in the cloud, where they utilized to improve dot product protocol and combined it into secure k-nearest neighbor's querying system. This technique also can counter-attack the assailant knowing the information that cloud data owner shares with query cloud client notwithstanding the encoded data [18]. Trapero et al. [19] have described sec Service Level Agreement (SLA) model created on defining a set of measurements for each security metric with which the cloud service providers can evaluate the strength of guaranteed service level objectives. For each service level objective, they have recognized a set of violations and detectable alerts, and they considered not only how to forecast, prevent, and eradicate secSLA violations, but also how to handle remedial and responsive activities in order to get-out producing more destruction, In the proposed secSLA trust factor is not included in the cloud consumers viewpoints. Kerberos is an authentication server whose main function is to authenticate client to server and server to client, the main security challenge in Kerberos is a key exchange between client and server. Talkhaby et al. proposed a new protocol with the powerful Diffie Hellman-DSA Key Exchange algorithm and the client's unique fingerprint impression tests. Biometric information utilized in the scheme comprehends the confinement of password-based authentication. In this work, using strong Diffie Hellman-DSA Key Exchange were performed as a Hard Key Exchange (HKE) mechanism with mutual authentication helped them to tackle the password predicting attack weakness in Kerberos version 5 protocols [20]. Nicanfar et al. proposed a Multilayer Consensus ECC-Based Password Authenticated Key-Exchange (MCEPAK) protocol, which allows secure communications between different layers of the smart grid control system and a home appliance. This method takes benefit of elliptic curve cryptography to provide a high-security level with a small key size while directing the possible resource limitations in the devices connected and they showed that proposed method decreases the security burden while giving elasticity against most of the attacks. Related to the X.1035 protocol, MCEPAK gains a lower load for the generation of the required passwords and hash function [21].

Hafizul et al. mentioned that ECC-based improves the security of key exchange of Peyravian-Zunic's secret key authentication method as well as expels the security defects Zhu et al.'s [22] method like clock synchronization issue and Identification and prevention of impersonation attack, etc. In this work, the proposed method enriches the computation of the symmetric key. It is used for private interchange of plaintext utilizing a symmetric key encryption algorithm. The performance examines of the proposed method is endorsed that the protection of data against impersonation attacks [23]. Murali et al. proposed a framework based on both traditional and quantum cryptography for cloud computing. This proposed method simplifies both the quantum channel and data channel for exchanging key and information correspondingly. The security framework based on quantum cryptography key distribution. The framework aims to support vital Quantum essential distribution communications with the cloud. This method is not suitable for the real cloud environment since the error rate is not an acceptable level of the cloud services providers [24].

Kumar et al. proposed a protocol that is based upon an RSA key distribution named as efficient centralized group key distribution protocol which is helpful from multiple points of view. It drastically decreases the computation load of the critical server by reducing the number of arithmetic operations needs to perform during crucial exchange and updating. However, the proposed method is not trusted for the cloud environment since it is considering the trust factor of cloud service providers [25].

Table 2.1 shows the merits and demerits of different key exchange protocol. Based on these the following conclusions have been made, the related works in the literature failed to incorporate certain factors such as Reliability and Recommendation which are identified as important aspects of key distribution in the hybrid cloud. We combined the features of ECC key exchange protocol using trust factor we introduced the method called Trust Factor Based Key Distribution Protocol (TFBKD); the proposed technique enables reduced computation and well-organized and secure group communication. It additionally ensured efficient rekeying for every communication session. Mehta and Singh talked about a few issues identified with exhibitions, security, and data transfer capacity issues in ASP .NET sites [26]. Singh et al. proposed technocrat ARIMA and SVR demonstrate (TASM), which gives a superior comprehension of the remaining burden and help to pick the estimating model. The gauging model additionally examined remaining testing for parameter setup of the models. Anyway, the proposed model works just in the web application situation [27]. Singh et al. present an Object-based Accountability Framework for data partaking in distributed computing. This structure provides responsibility to any client who is utilizing the cloud administrations utilizing a progressively particular and granular way utilizing article situated methodology. By utilizing this system, the client becomes more acquainted with that who can or who had gotten to his private data and when. This furnishes the client with a dependable choice to the client

Table 2.1
Comparison of the different key exchange protocol

Ref	Year	Technique	Advantage	Disadvantage
[11]	2018	Group key exchange protocol	High Efficiency	Protocol simply improved the two rounds of communications
[12]	2015	key exchange security	Efficiency and Reliability	Security provides only a single cloud service model Infrastructure as a service (IaaS)
[13]	2018	Blockchain-based security	Lower Transmission delay	It increases the computational load
[14]	2017	Secure Authentication Scheme (SAS)	High Efficiency	The Biometrics and password without prior communicating the trusted third party authority
[15][16]	2018	Biometric key authentication	Low communication cost	It prevents only replay attacks
[17]	2017	Crucial session based key exchange protocol	Efficiency and Reliability	Increases the computational load
[19]	2017	Service Level Agreement (SLA)	The strength of guaranteed service level objectives	The error rate is not an acceptable level of the cloud services cloud consumers
[20]	2016	Strong Diffie-Hellman Key Exchange	Efficiency and Reliability	Mutual authentication helped them to tackle for password predicting attack only.
[21]	2013	MCEPAK	Transmission delay	The trust factor is not included in the cloud consumers viewpoints.
[22][23]	2009 2013	Peyravian-Zunic's password authentication	Efficiency	Clock synchronization issue and Identification and prevention of impersonation attack
[24]	2016	Quantum cryptography key distribution	Low communication cost	The error rate is not an acceptable level of the cloud services providers
[25]	2018	Centralized group key distribution protocol	Efficiency and Reliability	The trust factor is not included in the cloud consumers's viewpoints.

to follow the exercises occurring over his private information [28].

3. Trust Factor Based Key Distribution Protocol. Cloud users submit a security demand(SD) to the resource broker to find the trust index (TI) of a cloud service providers, and resource brokers collect the truthfulness value from the cloud service providers, The resource Brokers map the task to the cloud service providers only if satisfies the following conditions ($TI >= SD$). The first difficulty in the group key communication is that interoperability between private cloud and public cloud. Data are passed across a multiple cloud deployment model. The second difficulty is establishing the trust relationship between the different cloud deployment models. To build a trust relationship between public cloud and private cloud, we introduced the method called Trust Factor Based Key Distribution Protocol (TFKD).

Our proposed method develops a trust factor based group key distribution security protocol to improve the security in the hybrid cloud deployment model. The security process is improved by considering the trust factors and ECC method. The proposed system mainly comprised of two stages namely, (i) Group Creation using the trust factor and (ii) Develop a hierarchy key distribution security protocol using ECC. These two stages are consecutively performed, and the security is increased more effectively and is discussed in section 3.1 and 3.2 respectively. The structure of our proposed Trust Factor Based Key Distribution Protocol using ECC method is illustrated in Fig. 3.1. We have designed a cloud service providers with the number of virtual machines communication nodes $C = \{c_1, c_2, c_3, ..., c_i\} 1 \leq i \leq N$ and each virtual machine communication node has varied number the jobs J. The virtual machine communication nodes jobs have the execution time also and it is represented as et. Moreover, the cloud service providers have the M number of resources, and it is denoted as $R = \{R_1, R_2, R_3, ..., R_i\} 1 \leq i \leq M$ and each resource has the processing time p_t. Before the grouping process, the virtual machine communications nodes want to allocate resources to complete their operations. During the

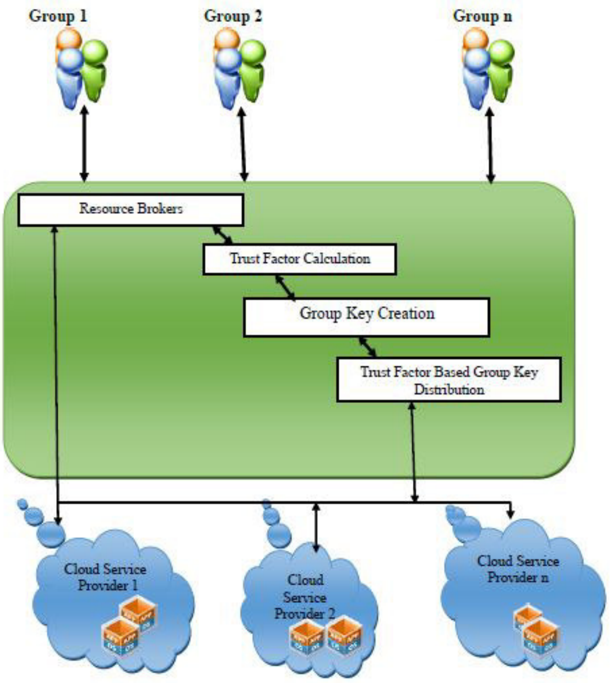

FIG. 3.1. *Structure of Trust Factor Based Key Distribution Protocol*

resource allocation, we have checked two conditions as stated below:

1. $J(e_t) < R_j(p_t)$
2. $J(e_t) > R_j(p_t)$

The jobs execution time, less than the corresponding allocated resource processing time means that we can avoid the job failure otherwise the job will be failed before it completes the operation. If condition (1) is satisfied, we can allocate the resources to the similar jobs. If the condition (2) is satisfied, we cannot do the resource allocation and go for other resources. Based on both the conditions (1) and (2) we allocate the resources to the jobs.

3.1. Group Creation Using Trust Factor. To perform the time-based clustering, we can create a database D, which contains joining time and the identification number of the communication node. Based on the time in the database, the virtual machine communication nodes are clustered. The clustered communication nodes are presented as $L = \{l_1, l_2, l_3, ..., l_g\}$ where g is a number of clustered groups and each cluster group l_g has a number of communication nodes.

$$I_g = \begin{cases} c_i, & c_i(j_t) > t_g \\ 0, & Otherwise \end{cases} \tag{3.1}$$

In Eq. 3.1, c_i is the virtual machine communication node, $c_i(j_t)$ is the joining time of the virtual machine communication node c_i and t_g is the time threshold value of the group g.

Trust factors: In cluster group lg a leader node is selected based on three trust factors. Here we have considered three trust factors namely: (RU), (RE), (RC).

1. Resource Utilization (RU): Resource utilization is calculated by checking the virtual machine communication node resource utilization time, i.e., which virtual machine communication node has utilized the maximum resource time.
2. Reliability (RE): Reliability factor is computed at different time slots, i.e. the virtual machine node which maintains the stable completion time at different slots.

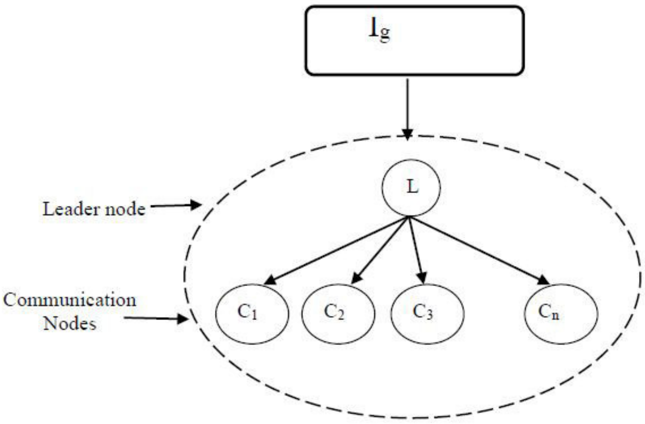

FIG. 3.2. *Cluster Model*

3. Recommendation (RC): the job completion time is calculated for all virtual machine communication nodes. The computed value is utilized as a node recommendation value.

Thus the created cluster group model with the group leader and members is shown in Fig. 3.2. The leader node is selected by averaging three trust factor values as,

$$a_i^{l_g} = \frac{RU_i^{l_g} + RE_i^{l_g} + RC_i^{l_g}}{3} \; i = 1 \; to \; n \tag{3.2}$$

where $RU_i^{l_g}$, $RE_i^{l_g}$ and $RC_i^{l_g}$ are the resource utilization, reliability, and recommendation of the virtual machine communication node, i from the cluster group l_g and $a_i^{l_g}$ represent the average value of the virtual machine communication node i. Based on these above mentioned three trust factors and the computed average value, a leader node is to be selected from each cluster group. Next, the secure communication process is performed within the group members, and key distribution process is also explained in the next section.

3.2. Developing Trust Factor Based Key Distribution Protocol using ECC. We have utilized an ECC method for creating a private and public key for the group members. Based on the created public key, the group key is also generated for all cluster groups by the leader. The generated group key is updated when the new group member joins or leaves from the cluster group. The group members private and public keys are produced by the ECC method makes communication among the group's members more secure and also the generated keys are robust.

3.2.1. Key Generation by ECC. We generate keys to the group members by the elliptic curve cryptography. Elliptic Curve Cryptography (ECC) is also called public key cryptography, where each user or the device participating in the communication usually has a couple of keys: a public key and a private key, and these keys are used do the cryptographic function. Small key size is the main advantage of ECC.

The function of the ECC method is defined over two finite fields: Binary field and Prime field. The field is selected with a finite number of points for cryptographic algorithm function. Here, we used prime field function by picking a prime number P, and finitely enormous numbers of points are created on the elliptic curve, such that the created points b are between 0 to K. Then, we have arbitrarily chosen one basic point $P_0(x_0, y_0)$ for key exchange function and this point satisfies the equation of the elliptic curve on a prime field, which is defined as

$$y^2 \; mod \; P = x^3 + ax + b \; mod \; P \tag{3.3}$$

In Eq. 3.3, a and b are the parameters that define the curve and x and y are the coordinate values of the generated point b. We randomly select one basic point P_0 that satisfies the above-mentioned Eq. 3.1. To complete the key exchange process, we need to select a private key P_v on the sender side, which is a randomly selected integer less than P and generate a public key $P_u = P_v * P_0$. Now each cluster group l_g members has individual private P_v and public keys P_u. By using these public keys, a group key is to be generated for each cluster group by the leader which is explained in the following subsection.

Group Key Creation: Secure communication is established within or between the group members we need a key identification and a secure key among the members. The group members are communicated with each other by their private P_v and public P_u keys and also exchange the information. Within the cluster group l_g, a group key is created for providing authentication between/within the group members. Thus the created group key is updated when any of the single members leaves or joins from or within the group and group members leave or join from or within the group.

Group Key: Each cluster group has the group leader and group members. The group key is created by using the group members public keys. The group key formation equation is stated as,

$$(G_k)^{l_g} = n^{l_g} \sum_{i=1}^{n} (p_{u_i})^{l_g} \tag{3.4}$$

where $(G_k)^{l_g}$ is denoted as group key of a cluster group l_g, n is the total number of communication nodes (members) in the group l_g and $(p_{u_i})^{l_g}$ is the public key of the i^{th} virtual machine communication node in cluster group l_g.

Updating Group Key: The created group key is updated when a new member joins or leaves from the group. The process of key updating is given below,
(i) A single member joins into the cluster group l_g it means the group key is updated by using the equation as described as,

$$(G_k)^{l_g}_{new} = (G_k)^{l_g} + (n * p_{u_{i+1}}) \tag{3.5}$$

(ii) When a single member leaves from the cluster group l_g it means the group key is updated by using the equation as stated below,

$$(G_k)^{l_g}_{new} = (G_k)^{l_g} - (n * p_{u_{i-1}}) \tag{3.6}$$

By using these, a group, private and public keys, the communication process is performed among group members. The public and the private keys created by the ECC technique make our proposed technique attain more secured communication and also the group key is used to authenticate the members within the groups. The Fig. 3.3 depict the Process of Trust Factor Based Key Distribution Protocol.

Encryption and Decryption: Subsequently the communication process is carried between the group members and the leader nodes by using the ECC technique. The leader node encrypts the messages by using ECC and sends that encrypted message to their group members.

The encrypted plain text is sent in the form of,

$$e = (M_e, C_j) \tag{3.7}$$

$$M_e = M_0 * p_0 \tag{3.8}$$

$$C_j = (x, y) + M_0 * (S(p_v) * p_0) \tag{3.9}$$

In Eq. 3.7, M_e is encrypted plain text which is computed by an Eq. 3.8 i.e., the multiplication of the original message M_0 with the basic point p_0 and C_j is evaluated by an Eq. 3.9. In Eq. 3.9, $S(p_v)$ is the private key of the sender. This message e is sent to the receiver. The receiver receives the message e and decrypts the message by the following formula,

$$d = M_e - (R(p_v) * C_j) \tag{3.10}$$

where $R(p_v)$ is the private key of the receiver.

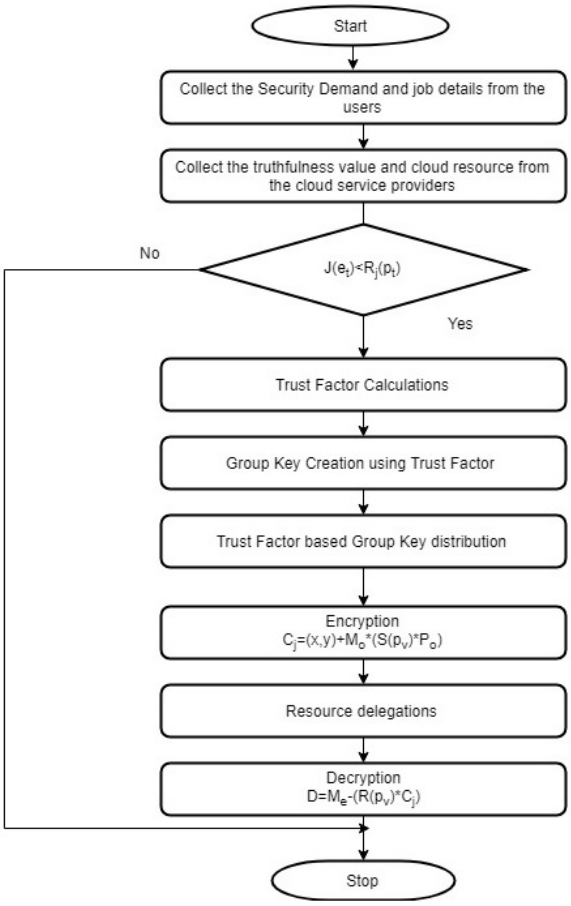
Start
Collect the Security Demand and job details from the users
Collect the truthfulness value and cloud resource from the cloud service providers
No
$J(e_t)<R_j(p_t)$
Yes
Trust Factor Calculations
Group Key Creation using Trust Factor
Trust Factor based Group Key distribution
Encryption
$C_j=(x,y)+M_o*(S(p_v)*P_o)$
Resource delegations
Decryption
$D=M_e-(R(p_v)*C_j)$
Stop

Fig. 3.3. *Process of Trust Factor Based Key Distribution Protocol*

3.3. Implementation of Trust Factor Based Key Distribution Protocol. The proposed protocol is implemented by using Open Nebula. Open Nebula is open source software for building cloud computing infrastructures and services. Open Nebula can be used to build public, hybrid and private cloud deployment model, and open nebula supports to integrate the existing cloud infrastructures [29] [30] [31]. Implementation of hardware and software details is listed in Table 3.1.

We have created two virtual datacenters using open nebula and public cloud deployment model using Amazon EC2. We integrated public cloud to existing open nebula private cloud environment which is depicted in Fig. 3.4. Users use the user interface which is written in Java and submit a job to the virtual machine present in the hybrid cloud. The submitted job is an interdependent job, and one job wants to communicate with another job running on the different virtual and different cloud environment model. If virtual machine securely intends to exchange the data between two different virtual machine, we need group key that should be transferred to all the virtual machine that is present in the hybrid cloud, so proposed Trust Factor Based Key Distribution Protocol is incorporated, and the job is communicated securely in the virtual data center. There are four crucial components in the implemented Hybrid Cloud deployment model such as Job Modeling Agent (JMA), Job Result status agent (JRSA), Trust server (TS) and Trust agent (TA) [32] [33].

Table 3.1
Implementation of hardware and software details

S.No	Hardware or Software	Details
1	Operating System	Supported host OS (CentOS, Ubuntu or RHEL) on all hosts
2	Hypervisor	Kernel-based Virtual Machine
3	Networking	CISCO SG-300 28 Port Managed Switch, The redundant 10Gbps storage network, Three 1Gb NIC interfaces ,1Gbps or 10Gbps network shared by service network, public and private virtual networks
4	Storage	IBM storage servers: iSCSI (40TB)
5	Authentication	X509
6	RAM	512 GB RAM
7	CPU	40 CPU Core (80 logical CPUs)
8	BUS	500-800MHz FSB
9	Amazon EC2 Public Cloud	#Instances Instance Type vCPU RAM(GB) OS 4 c3.2smal 8 4 Windows 1 m3.xlarge 4 15 Windows 2 c3.2xlarge 8 15 Windows 2 r3.2xlarge 8 32 Centos 1 r3.xlarge 4 32 Windows 1 m3.xlarge 4 15 Windows 2 m3.xlarge 4 15 Windows

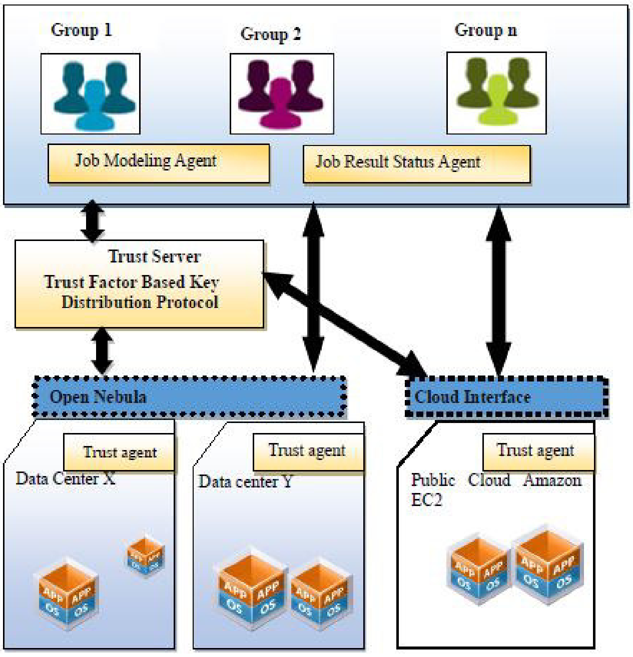

Fig. 3.4. *Integrated Key Distribution Protocol in Hybrid Cloud*

Job molding agent main function is to create a job; cloud consumers can specify the job completion time and the trust demand for their jobs. Their job can be scheduled for virtual machines according to their trust demand. Job result status agent obtains the results from the virtual machine. If the job execution is not completed in the virtual machines, then Job result status agent notify the failed job to the users by sending the failures notification message. A Trust server (TS) permits cloud consumers to maintain a trust relationship between

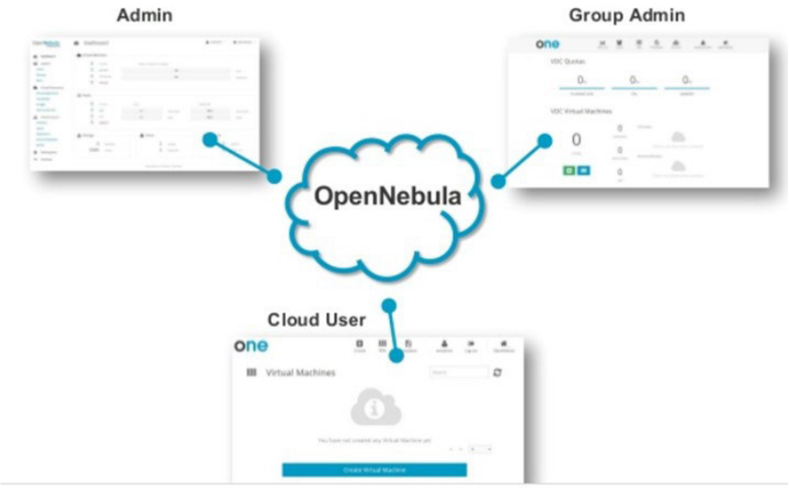

Fig. 3.5. *Open Nebula user account types*

the private cloud and public cloud to ensure cloud consumers can access virtual machines in the hybrid cloud. Trust agent collects the job success rate, virtual machine utilization, response capabilities, Reliability from the private cloud or public cloud and the transferred cumulative information to the Trust server to calculate the trust factor. If any virtual machine goes down, Trust agent generates the alarm signal and sent it to the Trust server for necessary action to be performed.

The Open Nebula and cloud interface permits a single point of entry to the hybrid cloud which consists of a trust agent and Datacenter. It secures authentication and authorization to the user. After authenticating the users private key, the server passes the job to the virtual machine communication node in the data center. Datacenter consists of Network Task Manager (NTM) and Target Method Interface (TMI). NTM receives job and starts to execute it. In the interim, NTM sends a response to JMA. After this, the cloud consumer can leave the system or surrender to another job. Throughout the execution, a Virtual space is formed to store all the information, including those wanted for execution and the files created by Abstract Job. NTM translates the Abstract Job into a collection of instructions that can be executed by Target Method Interface (TMI). For the duration of this conversion method, conversion tables in the incarnation database are used to map the abstract demonstrations to real instructions. TMI accepts incarnated job mechanisms from the NTM, and permits them to the confined batch schemes for execution. When the work is completed the virtual space is deleted.

The Open Nebula supports three types of account. All the account function is summarized below. The account interaction is illustrated in Fig. 3.5.
One admin: Special administrative account.
User: An OpenNebula user account.
Group: A group of Users.

4. Results and Discussion. The performance of the Trust Factor Based Key Distribution protocol is compared with the existing ECC and Diffie Hellman key exchange technique. In cloud hybrid cloud model; initially, we allocate the resources to each virtual machine communication nodes to complete their operations.

The resource allocation and their job completion performance are evaluated under three different number of communication nodes. The results that are obtained under these different job dataset sizes are given in Table 4.1.

After the resource allocation from the cloud system, the communication nodes are clustered by using their times. In our proposed method, we have taken the cluster limit as 8. The clustering result from our proposed technique is given in Table 4.2.

Table 4.1
Different Communication Nodes Job Completion Time with varying communication nodes

Virtual machine Communication nodes	Number of Jobs allocation respectively	Job completion time in seconds
10	4,5,10,10,8,8,10,4,6,5	38
20	2,2,6,4,2,2,5,1,9,3,3,3,4,1,2,5,4,5,6,2	18
30	1,3,3,2,1,5,2,3,4,3,4,3,1,4,1,4,3,2,3,1,1,1,3,1,2,3,4	13

Table 4.2
Time Based Clustering Results

Virtual machine Communication Nodes	Clustered Results			
	1	2	3	4
10	{2,3,7}	{4,8}	{9,1}	{5,6,10}
20	{12,13,3,6,8,1}	{11,7,2,15}	{4,9,14,18}	{5,10,17,20}
30	{14,21,27,28,11,8,13}	{0,4,12,17,20,24,15,23}	{3,6,10,19,22,25,29 }	{1,2,7,9,11,16}

In each cluster group, a leader node is selected based on the trust factors namely, (i) Resource utilization (RU) (ii) Recommendation (RE) and (iii) Reliability (RC). These three trust factors are calculated for all cluster group communication nodes. The sample calculated trust factors for the three cluster group results are given in Table 4.3.

Subsequently, in each cluster group, private keys and public key are generated by using the ECC technique whereas the group key is calculated by using these generated public keys and the private key of the group members. The security between the communications nodes is evaluated by creating random keys and measuring the security performances within the cluster group. The proposed and the existing methods error rate for the cluster groups is illustrated in above Table 4.4. The error rate is computed by the following formula:

$$ErrorRate = \frac{A_n}{T_n} X100 \tag{4.1}$$

where A_n is number of bits received and T_n is total number of bits in the original message. As it is seen from Fig. 4.1, our proposed technique has given a lower error rate than the ECC and Diffie Hellman algorithm. The existing algorithm obtains 40% average error rate performance in the communication within the cluster groups. Fig. 4.2 depict the rate of precision for man-in middle attack. The outcomes reveal that on categorization the information set with all the characteristics the average rate of error 89%, 83%, 79%, 66%,67%,97%,77% and 73% is acquired for ECC. Based on the usage of Diffe Hellman the error rate is increased. The error rate is decreased in Trust Factor Based Key Distribution Protocol is described during the usage of cluster formation with hybrid trust factor scheme.

From the Fig. 4.3, we can infer that key Renewal Computation time of proposed method provides 50 percentage lesser than the ECC and Diffie Hellman algorithm, this is because since the node is trusted, the computation virtual machine need not wait to assign new value while generating the key. Hence it is proved that our proposed technique provides more secure communication between the communication nodes than the existing method. Fig. 4.4 depict the Comparison of Key Renewal Computation time with different cluster size.

Fig. 4.5 depicts the comparison of Response time. The x-axis represents the different protocol and y-axis represents the response time in seconds. Response time is the length of time taken for a communication node to react to a given request or another communication node that is present in the different clusters. From Fig. 4.5, we can understand that the proposed method takes very less response time than ECC and Diffie Hellman protocol. This result is due to the management of the trust. The proposed approach enables efficient group communication.

Fig. 4.6 shows Resource utilization different method. The proposed method malicious node is not attacked by the trusted node, so it achieves the 91% resource utilization. In ECC and Diffie Hellman malicious node utilize the virtual machine resource this is due to ECC is not detecting the malicious node while they are

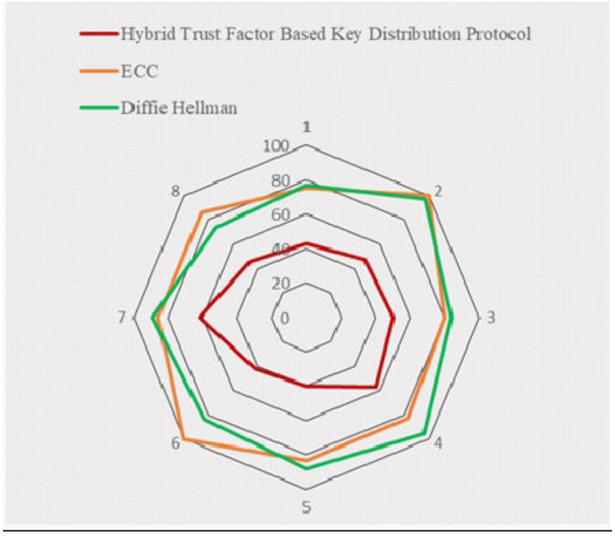

Fig. 4.1. *Radar chart Representation of the Proposed and Existing Methods Performance in terms of Error Rate*

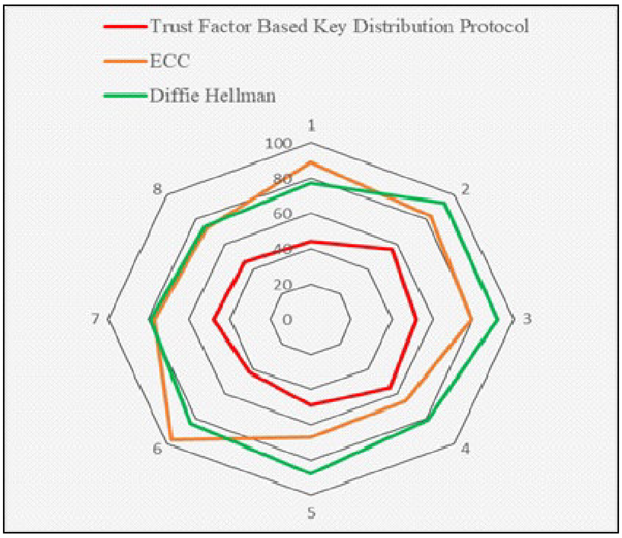

Fig. 4.2. *Error Rate for Man-in-Middle attack*

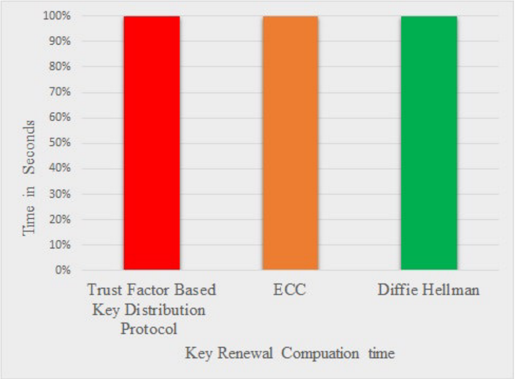

Fig. 4.3. *Comparison of Key Renewal Computation time*

Table 4.3
Trust Factors for Three Cluster Groups Communication Nodes and their Leader Nodes

Clusters	Virtual Machine Communication Nodes	Resource Utilization	Recommendation	Reliability
1	2	1	24	13.33
	3	0	34	9.88
	7	1	45	15.22
2	4	3	36	7.77
	8	3	57	10.33
3	9	0	52	11.44
	1	1	28	12.11
4	5	0	31	11.66
	6	3	39	18.0
	10	3	75	15.33

Table 4.4
Performance of Proposed and existing techniques Error Rate within the Cluster Groups

Cluster Groups	Error Rate (in %)		
	Trust Factor Based Key Distribution	ECC	Diffie Hellman
1	42.857	75	77
2	47.8	100	97
3	50	80	84
4	57	83.3	96
5	40	83.3	88
6	42.5	100	84
7	62	85.7	89
8	46	85.7	74

communicating with each other in the clusters.

Fig. 4.7 shows the communication overhead evaluation of Trust Factor Based Key Distribution Protocol with ECC and Diffie helman algorithm. Communication overhead time comparison, we consider that the user identity, virtual machine communication node identity, timestamp info, nonce , hash digest and Elliptic Curve Cryptography point $P = (P_x, P_y)$ are 256 bits, 128 bits, 64 bits, 32 bits correspondingly. Proposed method reduce commutation time between requester and virtual machine communication node compared to ECC and Diffie Hellman method. The reducing in the communication overhead time is suitable as the security is enhanced to an excessive point in the Trust Factor Based Key Distribution Protocol. The over-all investigational results confirm that the proposed protocol is efficient in terms of Communication overhead.

5. Conclusion and Future Work. We have proposed a Trust Factor Based Key Distribution protocol. The communication nodes were clustered using time-based clustering, and the leader was selected from among communication nodes by exploiting the trust factors. The ECC technique established the communication between the group members. The experimental results proved that our proposed Trust Factor Based Key Distribution protocol has reduced the error rate. Moreover, the comparative study shows that our proposed security protocols has given more secure communication and increase the resource utilization than the ECC and Diffie Hellman key exchange technique in the Hybrid cloud. Finally, the proposed method can be extended to allocate the job to the virtual machines by including defense capacity parameters such as intrusion detection, firewall, and antivirus capacity.

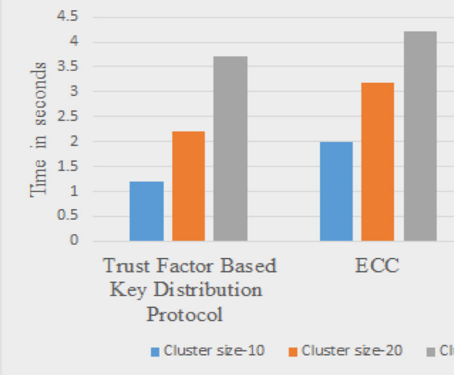

Fig. 4.4. *Comparison of Key Renewal Computation time with different cluster size*

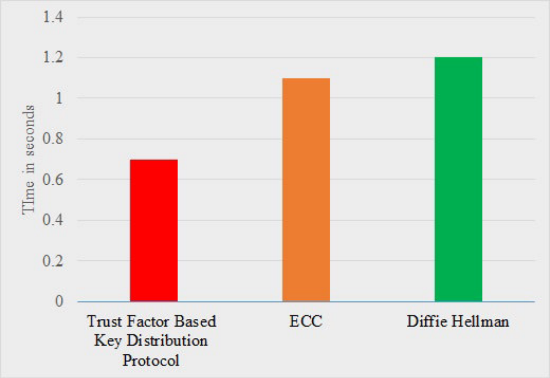

Fig. 4.5. *Comparison of Response Times*

REFERENCES

[1] Aleksandar Hudic, Paul Smith, and Edgar R Weippl. Security assurance assessment methodology for hybrid clouds. *Computers & Security*, 70:723–743, 2017.

[2] Ashish Singh and Kakali Chatterjee. Cloud security issues and challenges: A survey. *Journal of Network and Computer Applications*, 79:88–115, 2017.

[3] Shadi A Aljawarneh, Ali Alawneh, and Reem Jaradat. Cloud security engineering: Early stages of sdlc. *Future Generation Computer Systems*, 74:385–392, 2017.

[4] Gansen Zhao, Chunming Rong, Martin Gilje Jaatun, and Frode Eika Sandnes. Reference deployment models for eliminating user concerns on cloud security. *The Journal of Supercomputing*, 61(2):337–352, 2012.

[5] Mehedi Masud and M Shamim Hossain. Secure data-exchange protocol in a cloud-based collaborative health care environment. *Multimedia Tools and Applications*, pages 1–15, 2018.

[6] Antonio Celesti, Maria Fazio, Antonino Galletta, Lorenzo Carnevale, Jiafu Wan, and Massimo Villari. An approach for the secure management of hybrid cloud–edge environments. *Future Generation Computer Systems*, 90:1–19, 2019.

[7] Perumal Pandiaraja, Pandi Vijayakumar, Varadarajan Vijayakumar, and Raman Seshadhri. Computation efficient attribute based broadcast group key management for secure document access in public cloud. *J. Inf. Sci. Eng.*, 33(3):695–712, 2017.

[8] Flora Amato, Francesco Moscato, Vincenzo Moscato, and Francesco Colace. Improving security in cloud by formal modeling of iaas resources. *Future Generation Computer Systems*, 87:754–764, 2018.

[9] Maryam Farajzadeh Zanjani, Seyed Masoud Alavi Abhari, and Alexander G Chefranov. Group key exchange protocol based on diffie-hellman technique in ad-hoc network. In *Proceedings of the ACM 7th International Conference on Security of Information and Networks*, page 166, 2014.

[10] Y. Tan, J. Zheng, Q. Zhang, X. Zhang, Y. Li, and Q. Zhang. A specific-targeting asymmetric group key agreement for cloud computing. *Chinese Journal of Electronics*, 27(4):866–872, 2018.

[11] Fushan Wei, Neeraj Kumar, Debiao He, and Sang-Soo Yeo. A general compiler for password-authenticated group key exchange protocol in the standard model. *Discrete Applied Mathematics*, 241:78–86, 2018.

[12] Dan Gonzales, Jeremy M Kaplan, Evan Saltzman, Zev Winkelman, and Dulani Woods. Cloud-trusta security assessment

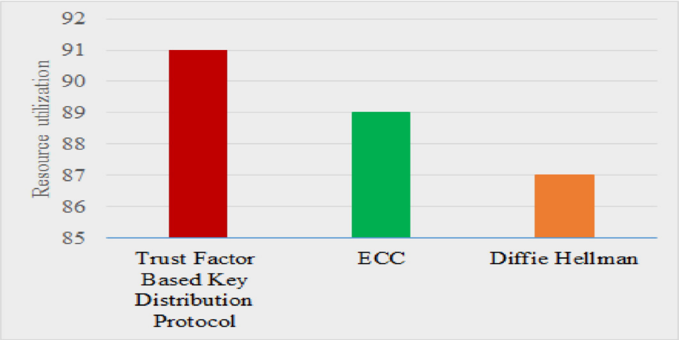

Fig. 4.6. *Comparison of Resource Utilization*

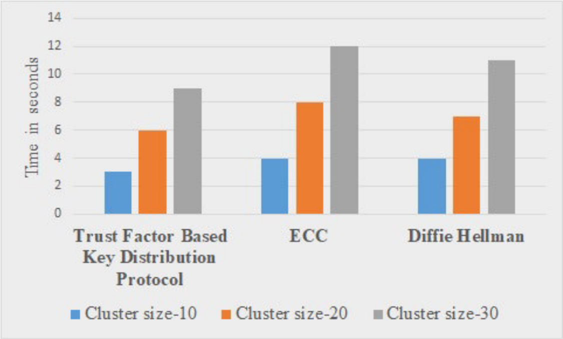

Fig. 4.7. *Comparison of Communication overhead with different cluster size*

model for infrastructure as a service (iaas) clouds. *IEEE Transactions on Cloud Computing*, 5(3):523–536, 2017.

[13] Jiaxing Li, Jigang Wu, and Long Chen. Block-secure: Blockchain based scheme for secure p2p cloud storage. *Information Sciences*, 465:219–231, 2018.

[14] Chia-Hui Liu and Yu-Fang Chung. Secure user authentication scheme for wireless healthcare sensor networks. *Computers & Electrical Engineering*, 59:250–261, 2017.

[15] Rajat Chaudhary, Anish Jindal, Gagangeet Singh Aujla, Neeraj Kumar, Ashok Kumar Das, and Neetesh Saxena. Lscsh: Lattice-based secure cryptosystem for smart healthcare in smart cities environment. *IEEE Communications Magazine*, 56(4):24–32, 2018.

[16] Sravani Challa, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, Saru Kumari, Muhammad Khurram Khan, and Athanasios V Vasilakos. An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Computers & Electrical Engineering*, 69:534–554, 2018.

[17] Subhas Barman, Samiran Chattopadhyay, Debasis Samanta, and Gaurang Panchal. A novel secure key-exchange protocol using biometrics of the sender and receiver. *Computers & Electrical Engineering*, 64:65–82, 2017.

[18] Lu Zhou, Youwen Zhu, and Aniello Castiglione. Efficient k-nn query over encrypted data in cloud with limited key-disclosure and offline data owner. *Computers & Security*, 69:84–96, 2017.

[19] Ruben Trapero, Jolanda Modic, Miha Stopar, Ahmed Taha, and Neeraj Suri. A novel approach to manage cloud security sla incidents. *Future Generation Computer Systems*, 72:193–205, 2017.

[20] Hamid Roomi Talkhaby and Reza Parsamehr. Cloud computing authentication using biometric-kerberos scheme based on strong diffi-hellman-dsa key exchange. In *Proceedings of IEEE International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 104–110, 2016.

[21] Hasen Nicanfar and Victor CM Leung. Multilayer consensus ecc-based password authenticated key-exchange (mcepak) protocol for smart grid system. *IEEE Transactions on Smart Grid*, 4(1):253–264, 2013.

[22] Lu Zhu, Sheng Yu, and Xing Zhang. Improvement upon mutual password authentication scheme. In *Proceedings of IEEE International Seminar on Business and Information Management*, volume 1, pages 400–403, 2008.

[23] SK Hafizul Islam and GP Biswas. Design of improved password authentication and update scheme based on elliptic curve cryptography. *Mathematical and Computer Modelling*, 57(11-12):2703–2717, 2013.

[24] G Murali and R Sivaram Prasad. Cloudqkdp: Quantum key distribution protocol for cloud computing. In *Proceedings of IEEE International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1–6, 2016.

[25] Vinod Kumar, Rajendra Kumar, and SK Pandey. A computationally efficient centralized group key distribution protocol for secure multicast communications based upon rsa public key cryptosystem. *Journal of King Saud University-Computer*

and Information Sciences, 2018.

[26] Sahil Mehta and Parminder Singh. An approach to security, performance and bandwidth issues in asp. net websites. *International Journal of Computer Applications*, 70(27), 2013.

[27] Parminder Singh, Pooja Gupta, and Kiran Jyoti. Tasm: technocrat arima and svr model for workload prediction of web applications in cloud. *Cluster Computing*, pages 1–15, 2018.

[28] Pradeep Singh, Parminder Singh, and Avinash Kaur. Object based accountability framework for information sharing in cloud computing. *International Journal of Computer Applications*, 115(19), 2015.

[29] C Yang, S Wang, and C Liao. On construction of cloud virtualization with integration of kvm and opennebula, 2011.

[30] Hyperconverged cloud architecture with opennebula and storpool, 2018.

[31] V Valliyappan and Parminder Singh. Hap: Protecting the apache hadoop clusters with hadoop authentication process using kerberos. In *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, pages 151–161. Springer, 2016.

[32] Anand Nayyar. Private virtual infrastructure (pvi) model for cloud computing. *International Journal of Software Engineering Research and Practices*, 1(1):10–14, 2011.

[33] Anand Nayyar. Interoperability of cloud computing with web services. *International Journal of ElectroComputational World & Knowledge Interface*, 1(1), 2011.

Edited by: Anand Nayyar
Received: Mar 16, 2019
Accepted: Apr 4, 2019