Computers, Materials & Continua
DOI:10.32604/cmc.2022.023554
Article

Tech Science Press

A Multi-Factor Authentication-Based Framework for Identity Management in Cloud Applications

Wael Said[1], Elsayed Mostafa[1,*], M. M. Hassan[1] and Ayman Mohamed Mostafa[2]

[1]Faculty of Computers and Informatics, Zagazig University, Zagazig, 44519, Egypt
[2]College of Computer and Information Sciences, Jouf University, Sakaka, 72314, Saudi Arabia
*Corresponding Author: Elsayed Mostafa. Email: sayedmostafa12529@gmail.com
Received: 12 September 2021; Accepted: 13 October 2021

Abstract: User's data is considered as a vital asset of several organizations. Migrating data to the cloud computing is not an easy decision for any organization due to the privacy and security concerns. Service providers must ensure that both data and applications that will be stored on the cloud should be protected in a secure environment. The data stored on the public cloud will be vulnerable to outside and inside attacks. This paper provides interactive multi-layer authentication frameworks for securing user identities on the cloud. Different access control policies are applied for verifying users on the cloud. A security mechanism is applied to the cloud application that includes user registration, granting user privileges, and generating user authentication factor. An intrusion detection system is embedded to the security mechanism to detect malicious users. The multi factor authentication, intrusion detection, and access control techniques can be used for ensuring the identity of the user. Finally, encryption techniques are used for protecting the data from being disclosed. Experimental results are carried out to verify the accuracy and efficiency of the proposed frameworks and mechanism. The results recorded high detection rate with low false positive alarms.

Keywords: Cloud computing; service providers; multi-layer authentication frameworks and access control

1 Introduction

Cloud computing is considered as one of the recent technologies in the evolution of the Internet. The cloud computing provides several services including computing power, servers, storage systems, applications, networks, and services. The cloud computing is considered as an advanced option over traditional methods regarding space, time, cost, and power. Three primary service models are applied on cloud computing to provide different levels of services to the users. These models are: platform as a service (PaaS), infrastructure as a service (IaaS) and software as a Service (SaaS). Each service model provides different layers of services such as databases, programming frameworks, operating systems, computing systems, networks, applications, and computing systems where the IaaS has the highest

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

capabilities on the cloud service providers. In PaaS, the user can control only deployed applications and cloud data while SaaS can control only limited user specific applications.

In cloud environment, the user can access to the cloud services using a set of authentication methods that can determine the user identity. After verifying the user identity, a set of authorizations will be allowed for each user according to his privileges and level of control [1]. The cloud services will be provided for each user based on the user IP address, access level, and domain of data [2]. Different security issues and challenges of cloud computing on organizations are identified [3]. These issues include data security, insider threats, denial of service, loss of control, and account hijacking. Each security threat has its potential harm on the cloud service provider and cloud data resources. Identity and access management (IAM) is considered one of the recent advances in securing cloud computing resources for creating and managing user identities. The cloud identity management can grant authorizations to different users based on their attributes and access roles.

This paper provides an enhanced feature for identity and access management (IAM) in order to secure user access on different cloud resources. The contribution of the paper is as follows:

- Providing an authentication procedure for checking users' identities in cloud applications.
- Applying a set of access control policies and roles that are embedded to the authentication procedure to verify user identity.
- Generating a user factor authentication method for verifying user privileges.
- Providing an intrusion detection method for verifying user activities and processes on cloud applications.
- Encrypting user data transactions on the cloud server for maintaining data confidentiality.
- Auditing user activities to trace access processes and user transactions on the cloud application data.

2 Related Work

Protecting and maintaining users' data privacy and confidentiality over cloud computing is considered as a challenging process [4]. As presented in [5], to improve the security of the authentication process, a dual-stage biometrics-based password authentication technique using smart cards was provided. The Master Server (MS) and the Authentication Server (AS) are two servers that work together to provide two-stage authentication (AS). The AS is in charge of authentication, while the MS is in charge of the remaining server. At the initial stage, elliptic curve cryptography-based ciphers are employed to create a connection and provide a secure protocol for communication between MS and AS across a secure channel.

As presented in [6], on the basis of trust between users and providers, to prevent a man-in-the-middle attack, encryption and decryption were used to safeguard user login by storing encrypted login IDs and passwords in the database. The user thumb impression was applied to authenticate user in case of password theft. The MD5 encryption was applied to secure the transfer user thumb impression over the network. As presented in [7], cognitive adaptive mechanisms and algorithms for enhancing system performance and increasing detection rate for internal intrusive users or administrators are provided. Low false positive alarms based on negative selection algorithm and danger theory are concluded. As presented in [8], elliptic curve cryptography algorithm was explained to offer better security model which serves with minimal computational power in cloud computing than another public key cryptosystem like RSA cryptosystem and also serves in secure data throw transmission. Cryptographic hash function was provided to generate a unique signature for user's plain text. The signature is used later for the authentication process of the user's plain text.

As presented in [9], two-server authentication architecture was provided Front-end Authentication Server (FAS) is a public server that exposes itself to users (or intruders), whereas Back-end Authentication Server (BAS) is a private server that stays behind the scenes. The (BAS) server's database has an up-to-date dictionary of all the client's passwords. Clients send authentication requests only to the public server in this scenario, yet both servers work together to complete the authentication operation. The authentication servers' failure-repair model was also supplied. As presented in [10], access control mechanisms along with current security threats was discussed and how the Access Control techniques based on authentication. Also, several security layers were provided. Also, cryptography and Socket & port-based programming were used to secure data from cryptanalyst and provide data transmission security.

As presented in [11], Pattern-Key Based Password Authentication was presented during the registration process, the user is given a 5×5 block grid with numbers ranging from 1 to 25 in major order. To register the selected pattern, the user first enters the corresponding location number from the grid. The user will also register a key for digits 0 to 9 in addition to the pattern. The key's function is to map the numbers selected from the grid's pattern to the key, resulting in a more secure password. A key converts numbers from 0 to 9 into any character, numeric, or special character. The user must next provide the number of dummy/fake values, known as Left Dummy and Right Dummy that must be entered before and after actual password values. As presented in [12], AES Encryption Algorithm was discussed to encrypt data before upload to the cloud also provides a unique data ID for every data item uploaded to the cloud. Also, key distribution center is used to store the result of XOR between encryption key and unique data ID. Also, message digest of the original message was generated to ensure data integrity.

As presented in [13], a registration and verification server were presented, which answers by issuing a smart card and providing a trust certificate to the Cloud service provider. After completing the registration process, the user submits a login request to the Cloud login server, which then passes it on to the Cloud authentication server. If the request is made for the first time, the authentication server sends it along with its key information to the registration and verification server for verification. Otherwise, it confirms the user's identification and establishes its authenticity with the user. The user uses the Cloud services until he or she signs out after the request for access is approved. As presented in [14], To provide a data protection paradigm, the Ciphertext-Policy Attribute Based Encryption (CP-ABE) access control model was introduced. It consists of four parts: user, authority, ciphertext storage space, and data access log. Also, a detection algorithm for data unauthorized access, which forms a closed-loop control, providing fast feedback evidence for the ongoing optimization of data access control strategies, and increases data protection integrity.

As presented in [15], The TH-KBBA Mechanism was presented, which combines the Tiger hash cryptographic algorithm and Kerberos Blowfish to improve authentication accuracy, confidentiality, and authentication time. For storing user information, the Tiger hash cryptographic function is used. User data is merged to produce a hash value with a predetermined size of 192 bits as an output. Authentication is done via the Kerberos Blowfish Biometric. Two servers are in charge of authentication. Ticket creation is used to verify authenticity. As presented in [16], The SK-AMFA methodology was presented that combines the suppression method and Schmidt-Samoa cryptography to enhance the privacy preserving rate of client sensitive data. In the registration phase, clients register their own information and store it in the CS, and in the authentication phase, they use multifactor authentication based on Schmidt-Samoa cryptography.

As presented in [17], a two-zone intrusion detection system with separate intrusion detectors for each tier was presented to protect the most crucial layers of the cloud (e.g., network layer, application layer). Signature-based detectors are installed in the first zone, whereas anomaly-based detectors are installed in the second zone. The first zone seeks to identify previously known attacks whereas the second zone seeks to target unidentified malicious events on the application layer. As presented in [18], a block chain-based distributed intrusion detection system and cloud computing infrastructure was used. Network intrusion detection systems (NIDS) and host intrusion detection systems (HIDS) are the two types of intrusion detection systems (HIDS). NIDS can be installed in the network backbone, gateways, switches, and server. Malicious actions are detected using HIDS. DIDS (distributed intrusion detection system) is also installed throughout the network. DIDS can make advantage of block chain. Because the centralized server is connected to many ids. As presented in [19], A Honeypot detection was presented as a method of caring data and resources in a cloud over Honeypot by applying it over an infrastructure application (Cloud Environment). Honeypot is a detection and security device which is required to be tested, assaulted, or bargained. It is used to detect and react, instead of taking corrective action. It can be applied in a cloud by either putting it before the firewall or after the firewall.

As presented in [20], Identity and Attribute based Cryptosystems were presented for Sharing Health records in Cloud Environment. IBBE scheme was used for the encryption of records at the user end and then stored on cloud, which can be decrypted later by authorized doctors. Also, a central authority is included which initializes the system, distributes the public parameters, provides secret keys to system users and keeps the master secret key. Patients can share their health records with doctors of their choice by encrypting them with IBBE, which feeds the doctors' identities, public parameters, and the patient's health data into the encryption algorithm. The data is subsequently uploaded to the cloud. Only authorized doctors, whose identities are revealed can decrypt the data using a secret key obtained from the central authority.

As presented in [21], A time and attribute-based dual access control and data integrity variable approach in cloud computing applications solves the problem of disclosure of private information caused by unfettered search by authorized users and incorrect data given by cloud servers was presented. First, in attribute-based encryption technology, a hierarchical time tree is developed, that restricts data search using time and attributes, preventing the leaking of private information caused by private key leakage. The Merkle tree and inverted index technology are then utilized to create a data variation tree, in which the cloud server's integrity of the search results may be checked without decryption. The problem of the cloud server returning data that is not accurate and full is thus resolved. As presented in [22], an approximation algorithm is used to solve content-based page using cloud VM Maximization problem in order to house many virtual machines that must contain minimum memory page transferring in each iteration. A heuristic algorithm to solve content-based page VM packing problem based on the approximation algorithm ratio is also presented. An access control model for securing page sharing in cloud virtual machines was presented in [23]. Security features, states, and transform rules are applied to adjust access control rules.

3 Methodology

As presented in Fig. 1, a framework is proposed for maintaining the identity of users, eliminating malicious participants' threats, and obtaining the privacy and confidentiality of data. The framework authenticates, controls and audits the user behavior with the interaction of intrusion detection mechanism. Also, advanced encryption standard (AES) algorithm is used as an encryption algorithm

to protect data during transmission to overcome the man-in-the-middle attack problem. This problem may compromise the transmission between application and DP and protect data at storage level in cloud database.

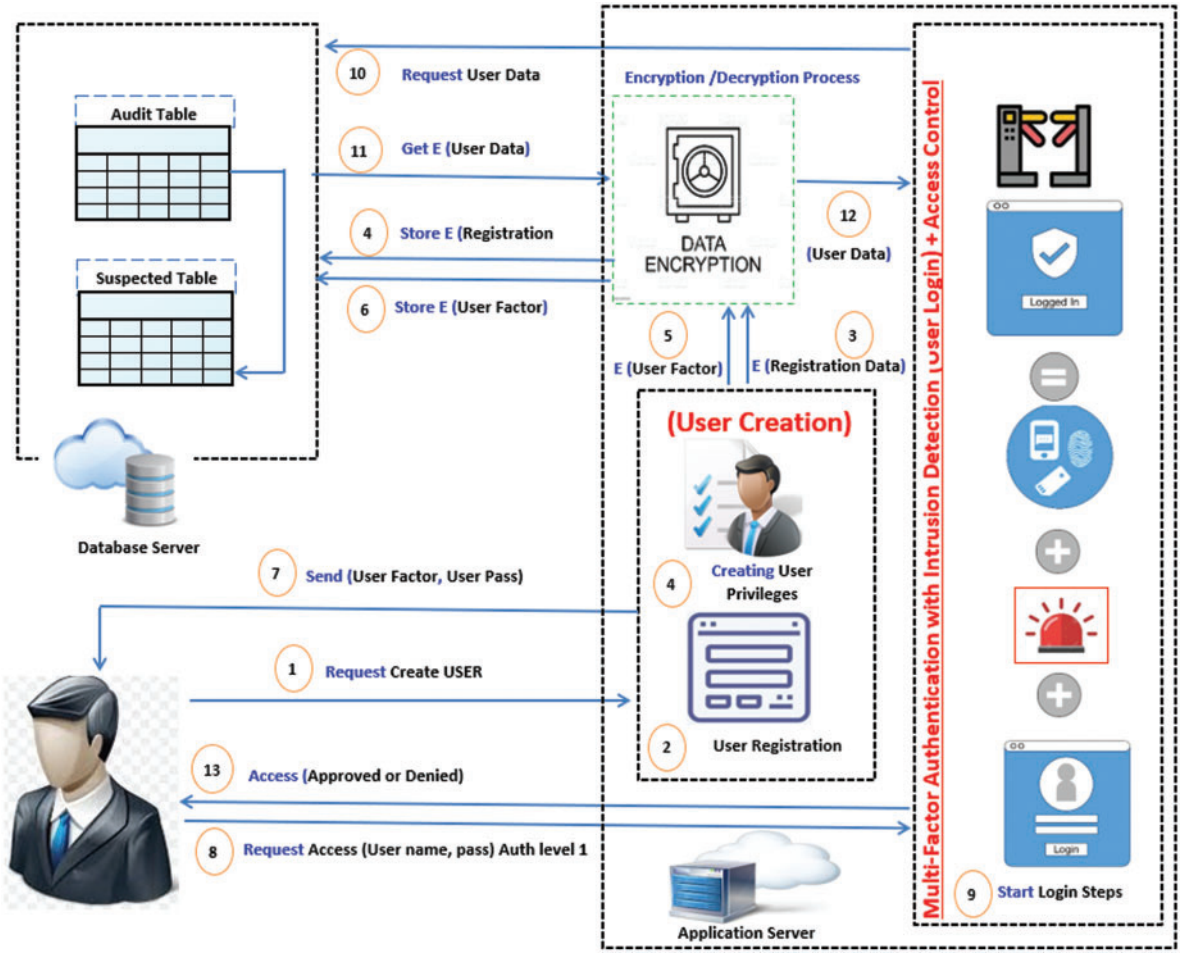

Figure 1: Proposed cloud security parameters

The overall framework is divided into three main phases as follows:

Phase I: User Creation Phase

In this phase the necessary procedures for user creation are initiated and the necessary attributes needed for a user to access the application are created. It is divided into two main stages, namely the User Registration and Creating User Privileges.

- User Registration: The user registers to the application using her/his secret attributes. Also, user must provide an emergency authentication method (mail, mobile, etc.).
- User Privileges Creation:

The assigned privileges to the user are executed at this stage. These privileges are given to the user by an application power user or application administrator. In addition, the access roles and user factor

(UF) are created based on the predefined user privileges. Once the user is created, her/his access class is defined based on the following classifications:

- Power user: the user who has high levels of authorizations and can grant/revoke any privileges to/from other users.
- Dept. Manager: is responsible for a group of users and has a full access to a specific database section.
- Normal User: a user who has limited access on application data for inserting, modifying, or deleting records.

Phase II: Granting Access Phase

In this phase the necessary procedures for granting a user to path throw and access the application are determined. It depends on main four components. These components are the Multi-Factor Authentication, intrusion detection audit table, and suspected table.

- Multi-Factor Authentication: By granting the user to access the application, the user must path throw multi levels of authentication mechanism that is based on three levels. These levels are:
  - Level 1: user name and password of the authorized user.
  - Level 2: the user authorized factor that must be issued to the application system. The system transfers the factor to the intrusion detection step that works as a multi-level of authentication to check the user factor and raise an alarm when suspicious actions of users are detected. The intrusion detection step is a continuous step in the following phases. The IDS step passes the user through different procedures in order not to raise false alarms for authorized users. The overall IDS procedures will be explained in more detail in Section 4.
  - Level 3: a user certificate is created after verifying the user in level 2.
- Audit Table: each executed transaction starting from login phase and access control is recorded into the audit table. The audit table records all user activities that have been executed on the application data. The audit table summarizes all raised alarms for the users in order to increase future countermeasures
- Suspected Table: this table is created to record all suspicious users that have violated their privileges.

Phase III: Encryption and Decryption Phase

Once the user submitted her/his data on the application, an encryption process is executed using AES algorithm. The encrypted data are stored in the database server while the primary key of the recorded data is not encrypted in order to retrieve the data more easily. When the user requests data from the application, a decryption process is executed to retrieve the data from the database server. Data is stored on cloud databased and transferred to the application during encryption and decryption processes. The AES encryption algorithm is used to secure data storage with less memory consumption and less computational time. The AES algorithm provides high security countermeasure from unauthorized intruders [24]. It is also used to encrypt large data files by improving the encryption efficiency. The privacy of user's data is maintained on the cloud platform that can resist attacks on the plaintext while meeting the needs of encoding and decoding speed [25]. The AES algorithm is considered as an effective algorithm in data storage and transmission for approximately 29% of cloud data centers at the end of 2023 [26].

4 Proposed Frameworks

The proposed frameworks are used to provide a complete security procedure based on four interactive steps. These steps start from user registration, creating and applying user privileges, performing authentication process, and finally applies intrusion detection mechanism for tracing and monitoring any deviations on the cloud services. These steps are explained in the following subsections.

4.1 User Registration

As presented in Fig. 1 step 2, the $user_k$ requests access to the application server to use the cloud resources. The parameters of $user_k$ are sent to the application server and then are encrypted using AES 128 cryptographic algorithm. The encrypted data are stored on the cloud database server for the future login processes, the **User Registration** process is presented in Fig. 2.

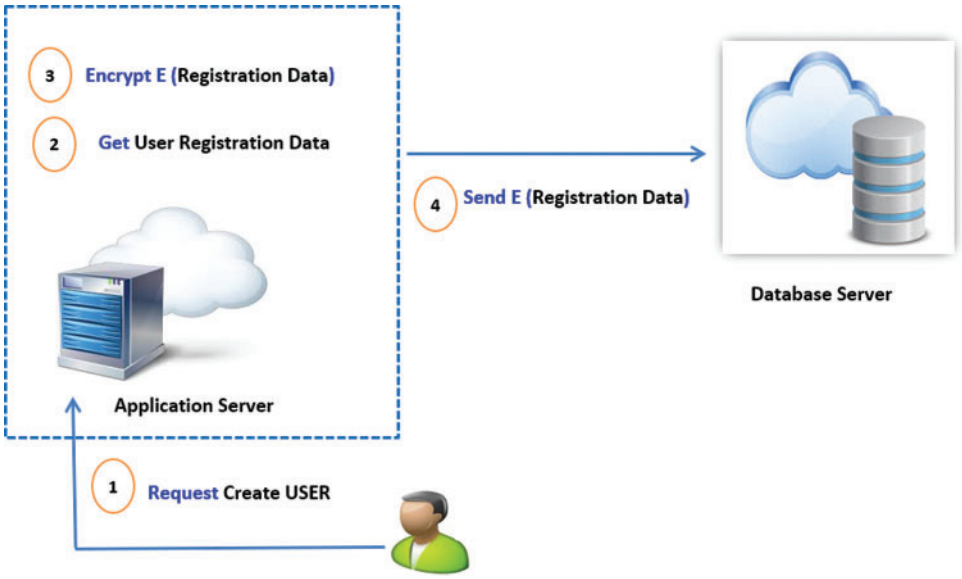

Figure 2: User registration phase

4.2 User Privileges Creation

The process of creating user privileges and authentication factors as presented in Fig. 1 step 4 is explained in Fig. 3 as the application server requests the parameters of $user_k$ from the cloud database server. The encrypted parameters of $user_k$ are sent to the applications server. Once the encrypted parameters are received, the applications server performs a decryption process to start creating $user_k$ privileges. The user privilege process contains three major steps. Firstly: the $user_k$ access class that identifies the classification of user activity on the cloud services. Secondly: the access authorization roles are defined based on the $user_k$ access class. The access roles are encrypted and sent to be stored on the cloud database server. Finally: a unique user factor will be created based on the predefined authorization roles to verify the user during the login process. The defined user factor is encrypted and sent to the cloud database server.

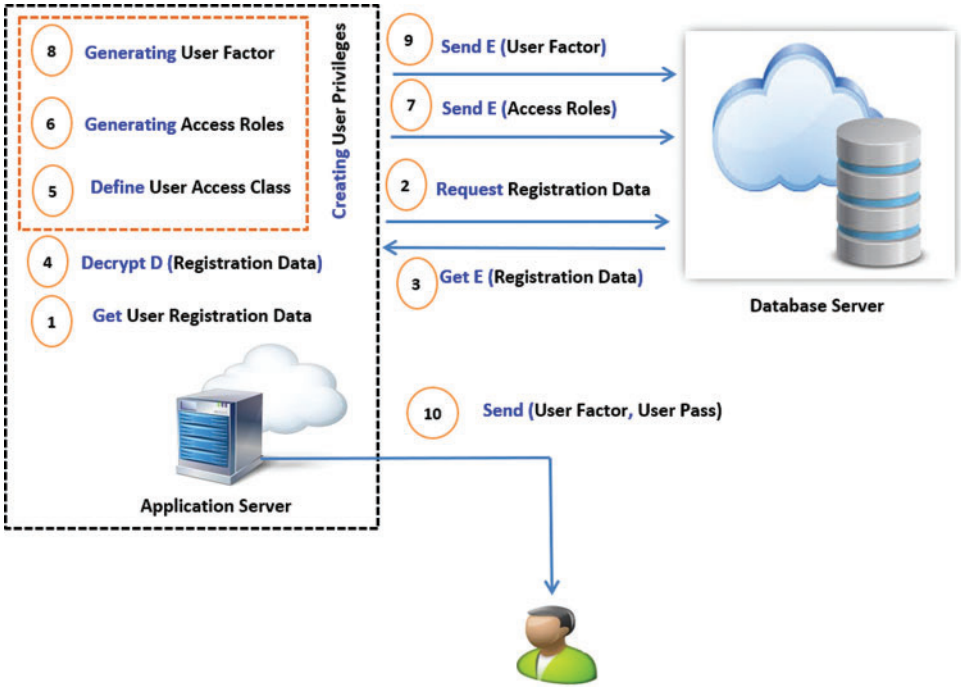

Figure 3: User privileges creation

As discussed before, the user privilege creation is based on three major steps to verify uniquely identify the user on the cloud application server. These steps are defined in the following subsections.

4.2.1 User Access Class

As indicated in Fig. 3 step 5, the access class for each user is defined based on her/his privileges and authorizations on the cloud application. As presented in Tab. 1, the power user has a full access on the cloud application resources for performing DML and other database processes. The department manager has limited access over the cloud application resources based on her/his predefined authorizations. Finally, the normal user has specific privileges less than other user classes on cloud applications.

Table 1: User access class

User authentication privileges	User access classes		
	Power user	Dept. manager	Normal user
Search all data	√		
Search specific departmen data	√	√	√
Insert	√	√	√
Update	√	√	√

(Continued)

Table 1: Continued

User authentication privileges	User access classes		
	Power user	Dept. manager	Normal user
Delete	√	√	√
Access all times	√		
Access specific times	√	√	√
Grant privileges to another user	√	√	
Create user	√	√	
Delete user	√		
Set user inactive	√	√	
Create Factor	√	√	
Print reports	√	√	√

4.2.2 Generating Access Roles

As presented in Fig. 3 step 6, the access roles are automatically generated depending on the user access class step before generating a new role the responsibilities assigned to that user which making the role are compared with the existing roles responsibilities if they matched an existing role the role code is assigned to the user or generating a new role as shown in Fig. 4.

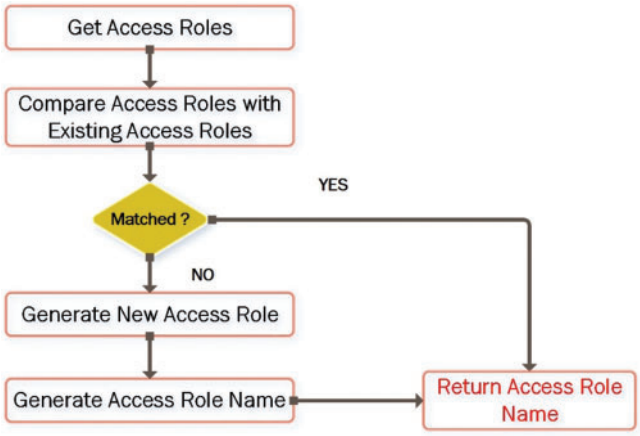

Figure 4: Generating access roles

4.2.3 Generating User Authentication Factor

The user factor plays a vital role in the authentication and intrusion detection processes. It is considered as the second level of authentication during the intrusion detection mechanism that checks the unique factor characteristics (fixed length of 10 character and an expiry date) for ensuring user identity. Based on the user authorizations and access class, each user has granted and denied privileges, see Fig. 3 step 8. As presented in [6], each granted process will have a value of 1 while the denied process will have a value of 0. In this paper, a wide range of user authorizations over cloud applications have

been applied such that each granted cloud process will have a value of 1 and the denied cloud process will have a value of 0. As a result, each user will have a certain user factor for his predefined privileges. Even if, different users have the same privileges, the user factor will be different due to the following mechanism:

$$user_p = \begin{cases} 1, & granted\ privilege \\ 0, & denied\ privilege \end{cases}$$
$$Random_x = assign\ 2\ deciaml\ numbers$$
$$Binary_x = Convert\ (Random_x)_2 88$$
$$Final\ user\ factor\ \ user_f = user_p\ ||\ Binary_x$$
$$Encrypt\ \ = E\ (user_f,\ K)$$
$$Overall\ user_{Auth\,f} = \forall\ x \in user_f\ such\ that\ x_i\ \subset\ \{x_1,\ x_2\ ,\ \ldots, x_n\}$$
$$Login\ user_{Auth\,f} = concat\ \{x_1,\ \ldots\ , x_5||\ x_{n-4}, \ldots, x_n\}$$

As explained in the previous formulas, each user authentication factor $user_f$ is created by combining the user privileges with a two-digit random number. The random number is converted to binary and then combined with the user privileges to create the final user factor $user_f$. The $user_f$ is encrypted using AES-128 encryption algorithm to create the overall user authentication factor $User_{Auth\,f}$. The user login authentication factor is then created by combining the first digits $\{x_1,\ \ldots\ , x_5\}$ with the last digits $\{x_{n-4}, \ldots,\ x_n\}$. For example:

- 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1
- two random decimal numbers are generated.
- Then they converted into binary values.
- Then the value encrypted Using AES (71C71D16AD94DC3E2FD79AD53EF47F9A)
- Then only first five and last five characters sent to the user (71C7147F9A).

5 Proposed Access Mechanism

Assume $user_k$ wants to access the application, s/he had to path throw the login steps which is consists of multi-level authentication with the interaction with intrusion detection, Fig. 1 step 9. The login scenario steps are gone as follows:

5.1 Multi-Level Authentication

Level 1

In the first level of authentication, the $user_k$ enters the user's name and password as shown in Fig. 5 step 1. A request is sent to database server to get the access roles for the entered user name after checking the user's name and password the encrypted access roles is sent to the application server as shown in Fig. 5 step 1 and step 2. The access roles are decrypted and be checked, Fig. 5 step 6 and step 7.

Level 2

After checking the user's name, password and access roles, a request is sent to the $user_k$ to get the user access factor as shown in Fig. 5 step 6 and step 7. A request is sent to database server to get the $user_k$ access factor the encrypted user access factor is sent to the application server (Fig. 5 step 8 and step 9). Then the user access factor is decrypted (Fig. 5 step 10). Then the $user_k$ access factor which is entered in step 7 is checked throw the intrusion detection steps (check factor length, check factor

validity, compare the entered factor value with factor in Fig. 5 step 10 finally check suspected table) as shown in Fig. 5 step 11.

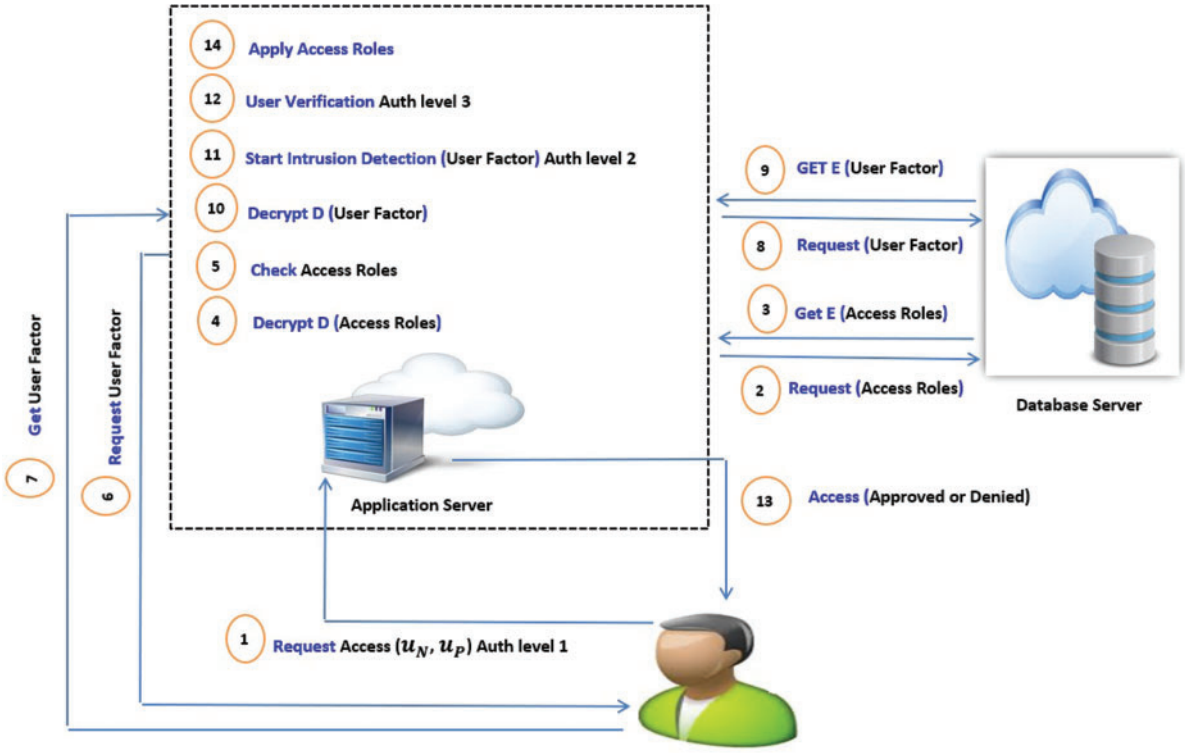

Figure 5: Overall access mechanism

Level 3

The last step in authentication is to send a verification message to the $user_k$ to be sure before allowing to access the application, Fig. 5 step 12.

The overall access steps and intrusion detection processes are explained in Fig. 6.

5.2 Intrusion Detection Steps

1. check factor length.
   In the proposed scheme the authentication factor length is 10 characters to ensure that the user is not an intruder he cannot enter a factor of a different length.
2. check factor validity.
   The user authentication factor has a valid period (start date and end date) the user during the login cannot uses an expired factor.

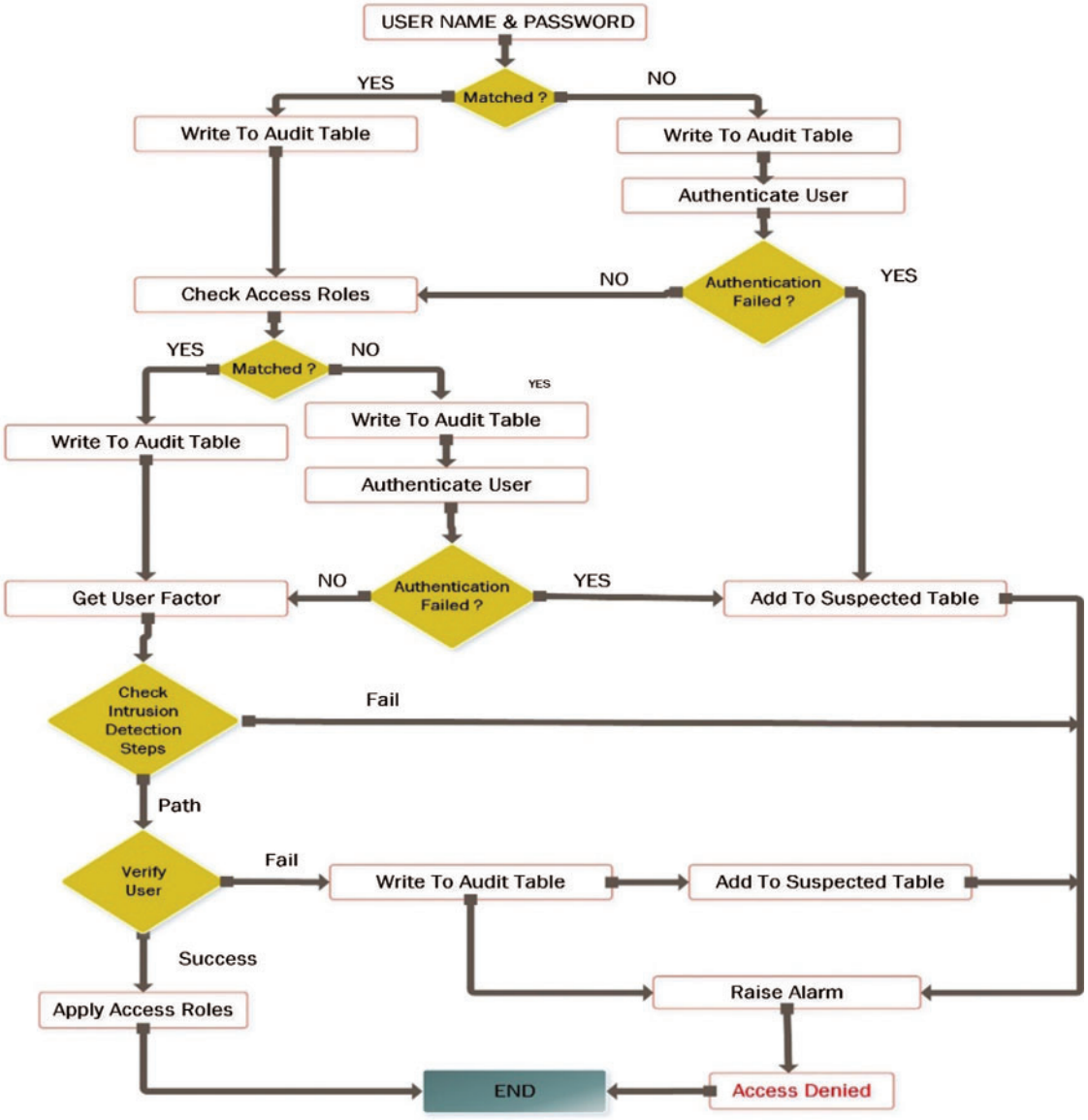

Figure 6: Overall access steps with intrusion detection

3. compare the entered factor value with factor.
   After ensuring that the entered factor is valid and its length is ten then it be compared with the factor of the user that was previously stored in the database server.
4. finally check suspected table.

Before allowing the user to access the application the suspected table must be checked to ensure that the user does not exist in it.

The overall intrusion detection steps are explained in Fig. 7.

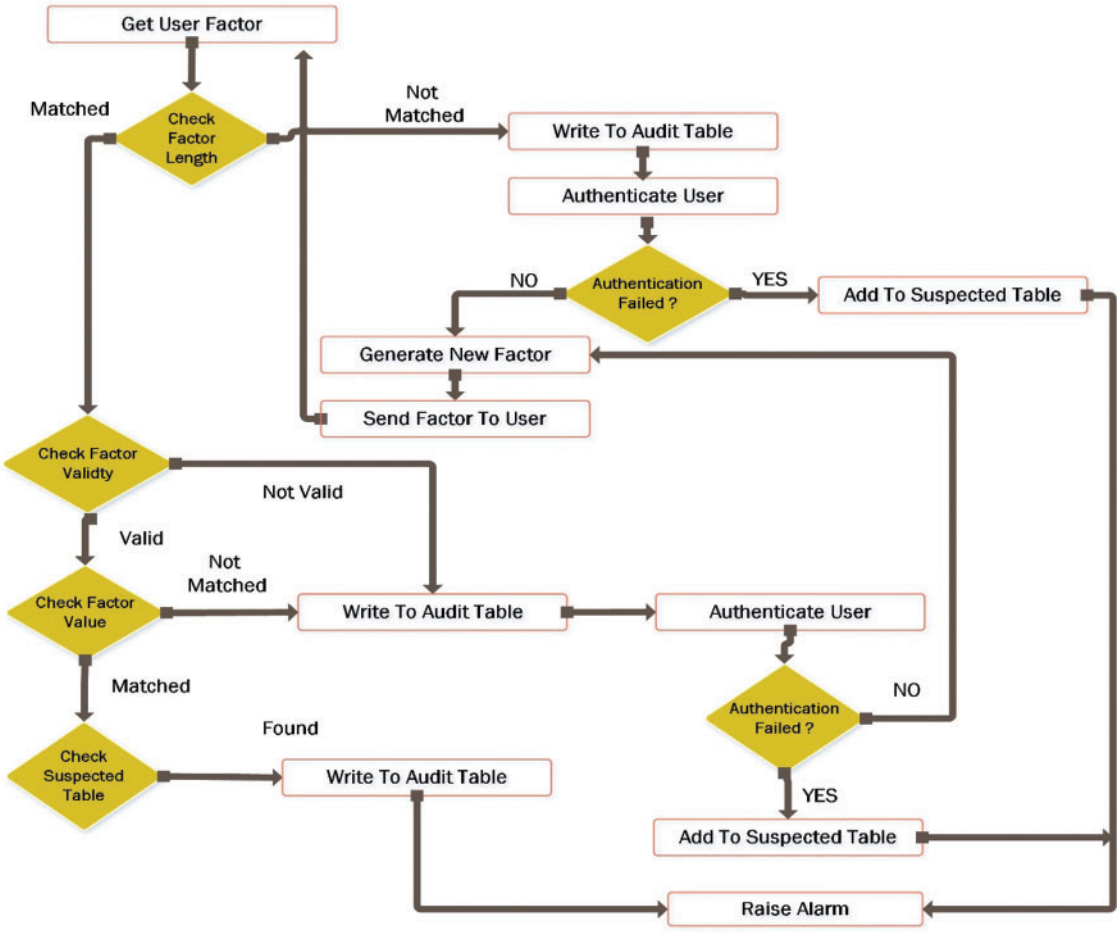

Figure 7: Intrusion detection steps

6 Experimental Results

In this section, a dataset was applied as presented in Tab. 2 to evaluate the efficiency of the proposed access procedures which consists of access roles, multi-level authentication with the cooperation of intrusion detection and also the additional authentication process ($AUTH_{ADD}$) was added to enhance the efficiency. The efficiency of the proposed access procedures is determined by measuring the overall DR, FN alarm rate and FP alarm rates. Tab. 2 provides a description of the dataset that is used in our experimental results and a classification of the users' types whether they are normal users or intruders.

The False Positive (FP) alarms allude to the possibility of normal users being wrongly identified as harmful users. This is shown in Formula (1).

$$FP = \frac{N_f}{N} \times 100\% \tag{1}$$

As shown in Fig. 8, without using ($AUTH_{ADD}$) the percentage of the normal user who are considered intruder and added to the suspected table is 2.3%. We assume the users are normal user who try to access the application some of them may forget password or the factor so the ($AUTH_{ADD}$)

process was merged to the login steps to reduce the FP rate. After using the ($AUTH_{ADD}$), the FP rate is decreased and records 0.5% but also, we still have FP rate according the ($AUTH_{ADD}$) failure or the user verification failed or the user entered a wrong code during the ($AUTH_{ADD}$) or verification phase.

Table 2: Experimental details

Normal users		Intruders			
3000		900			
Without authentication	With authentication	Without authentication		With authentication	
70 users (2.3%)	15 users (0.5%)	Insider 18 Users (2%)	Outsider 0 Users (0%)	Insider 18 Users (2%)	Outsider 9 Users (1%)
FP rate		DR & FN Rates			

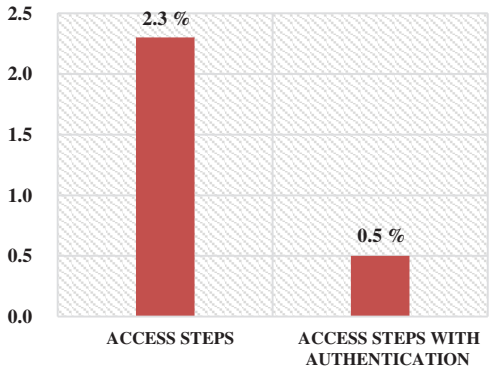

Figure 8: FP alarms

Detection rate (DR) shows the ratio of detected intrusion actions by dividing the whole number of detected users (Nd) by the whole number of users (N). This is shown in Formula (2).

$$DR = \frac{N_d}{N} \times 100\% \tag{2}$$

As shown in Fig. 9, Based on using a multilevel authentication and intrusion detection the detection rate makes high percentages 98% but we notice the merge between the ($AUTH_{ADD}$) and access steps may help an intruder to access the application that decreased the DR to 97%.

The False Negative (FN) alarms specify the number of malignant users who pass the login procedures. The ratio is calculated by dividing the number of passing malignant users (Np) by the whole number of examined users (N). it is shown in formula (3).

$$FN = \frac{N_p}{N} \times 100\% \tag{3}$$

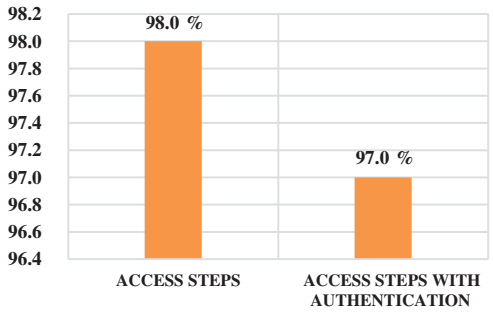

Figure 9: Overall DR

As shown in Fig. 10, FN records low values. The remaining percent 2.0% of the FN comes from insider intruders who can already access the application as normal users and try to do malicious activates without using the ($AUTH_{ADD}$) only the insider intruder can access the application. after using the ($AUTH_{ADD}$) an outsider intruder may use it to access application so the percent of the FN increased to 3.0%. Assume an outsider intruder knows the user's name and steal the authentication email of a user every time he enters a wrong password or wrong factor, he fires the ($AUTH_{ADD}$) which sends the required access information to the intruder, so after studying the audit table we found that a user should not use the ($AUTH_{ADD}$) more than two times.

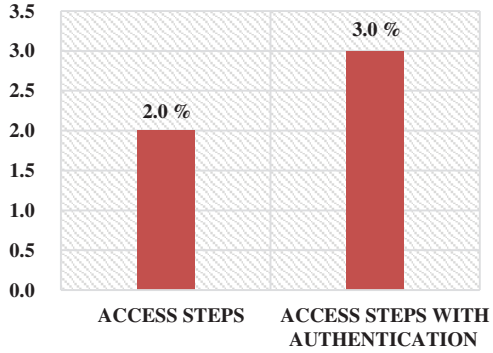

Figure 10: FN alarms

Also, we have insider intruders already have the login parameters and can access the application and try to search unauthorized data here comes the role of applying access roles which make the user just sees the data assigned to his department and according to his class.

Also, we have the insider intruders who try to show or steal data assume a user steals the factor and authentication email of his manager and uses the two times ($AUTH_{ADD}$) he can access the data, so that the user factor has expiry date according to the importance of the data and the user class. so, after studying the audit table we found that the user that always using the ($AUTH_{ADD}$) process to access the application must be blocked until making sure of his unnormal activities.

7 Conclusion and Future Work

In platform as a service (PAAS), the customer is responsible for both data and the application. Securing the application and data is based on the responsibility of the consumer. The consumer must provide the application with advanced techniques. Ensuring the identity of the user who can access the application is based on registering and authorizing access rights in the user creation phase, and then in the login phase for identifying, authenticating and controlling user to have access to applications. Different security frameworks and mechanism are applied to secure user confidential data based on access control policies. Experimental results are conducted to measure the accuracy and efficiency of the proposed frameworks and mechanism and proved that the mechanism achieved high detection rates with low false positive alarms.

Adding a second level of intrusion detection applied after the user access to the application. The second level of intrusion detection based on studying user behavior. Also enhancing the authentication process by adding a biometric authentication technique to reduce the levels of authentication used.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors state that they have no conflicts of interest to report regarding the present study.

References

[1] S. Logesswari, S. Jayanthi, D. KalaSelvi and V. Aswin, "A study on cloud computing challenges and its mitigations," *Materials Today: Proc.*, vol. 1, pp. 1–5, 2020.

[2] L. Alhenaki, A. Alwatban, B. Alamri and N. Alarifi, "A survey on the security of cloud computing," in *2019 2nd Int. Conf. on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, pp. 1–7, 2019.

[3] L. B. Bhajantri and T. Mujawar, "A survey of cloud computing security challenges, issues and their countermeasures," in *2019 Third Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, pp. 376–380, 2019.

[4] Y. Shi, "Data security and privacy protection in public cloud," in *2018 IEEE Int. Conf. on Big Data (Big Data)*, Seattle, WA, USA, pp. 4812–4819, 2018.

[5] M. Boopathi and M. Aramudhan, "Secure server-server communication for dual stage biometrics – based password authentication scheme," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 819–829, 2018.

[6] S. Ojha and V. Rajput, "AES and MD5 based secure authentication in cloud computing," in *Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, pp. 856–860, 2017.

[7] W. Said and A. M. Mostafa, "Towards a hybrid immune algorithm based on danger theory for database security," *IEEE Access*, vol. 8, pp. 145332–145362, 2020.

[8] M. Chakraborty, B. Jana and T. Mandal, "A secure cloud computing authentication using cryptography," in *Int. Conf. on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR)*, Ernakulam, India, pp. 1–4, 2018.

[9] D. Chattaraj and M. Sarma, "Dependability quantification of cloud-centric authentication frameworks," in *IEEE 11th Int. Conf. on Cloud Computing (CLOUD)*, San Francisco, CA, USA, pp. 840–844, 2018.

[10] N. C. Ravi, M. N. Babu, R. Sridevi, V. K. Prasad, A. Govardhan *et al.*, "Inspecting access controls in cloud based web application, "*Second Int. Conf. on Computing Methodologies and Communication (ICCMC)*, Erode, India, pp. 262–269, 2018.

[11] M. H. Zaki, A. Husain, M. S. Umar and M. H. Khan, "Secure pattern-key based password authentication scheme," in *2017 Int. Conf. on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, Aligarh, India, pp. 171–174, 2017.

[12] N. Mishra, T. K. Sharma, V. Sharma and V. Vimal, "Secure framework for data security in cloud computing," *Soft Computing: Theories and Applications*, Singapore, Springer, pp. 61–71, 2018.
[13] B. B. Gupta and M. Quamara, "An identity based access control and mutual authentication framework for distributed cloud computing services in IoT environment using smart cards," *Procedia Computer Science*, vol. 132, pp. 189–197, 2018.
[14] H. Zhang, F. Lou, H. Wang and Z. Tian, "Research on data protection based on encrypted attribute access control in cloud computing," *5th Int. Conf. on Information Science and Control Engineering (ICISCE)*, Zhengzhou, China, pp. 450–453, 2018.
[15] K. M. Prabha and P. V. Saraswathi, "Tiger hash kerberos biometric blowfish user authentication for secured data access in cloud," in *2nd Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, pp. 145–151, 2018.
[16] K. Mohana Prabha and P. Vidhya Saraswathi, "Suppressed K-anonymity multi-factor authentication based Schmidt-Samoa cryptography for privacy preserved data access in cloud computing," *Computer Communications*, vol. 158, pp. 85–94, 2020.
[17] M. Jelidi, A. Ghourabi and K. Gasmi, "A hybrid intrusion detection system for cloud computing environments, "*Int. Conf. on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, pp. 1–6, 2019.
[18] M. Kumar and A. K. Singh, "Distributed intrusion detection system using blockchain and cloud computing infrastructure," *4th Int. Conf. on Trends in Electronics and Informatics (ICOEI) (48184)*, Tirunelveli, India, pp. 248–252, 2020.
[19] P. S. Negi, A. Garg and R. Lal, "Intrusion detection and prevention using honeypot network for cloud security," in *10th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, pp. 129–132, 2020.
[20] P. K. Maganti and P. M. Chouragade, "Secure application for sharing health records using identity and attribute based cryptosystems in cloud environment," in *3rd Int. Conf. on Trends in Electronics and Informatics(ICOEI)*, Tirunelveli, India, pp. 220–223, 2019.
[21] Q. Zhang, S. Wang, D. Zhang, J. Wang and Y. Zhang, "Time and attribute based dual access control and data integrity verifiable scheme in cloud computing applications," *IEEE Access*, vol. 7, pp. 137594–137607, 2019.
[22] H. X. Li, W. J. Li, S. G. Zhang, H. D. Wang and Y. Pan, "Page-sharing-based virtual machine packing with multi-resource constraints to reduce network traffic in migration for clouds," *Future Generation Computer Systems-the International Journal of Escience*, vol. 96, pp. 462–471, 2019.
[23] Z. Tang, X. F. Ding, Y. Zhong, L. Yang and K. Q. Li, "A self-adaptive bell-lapadula model based on model training with historical access logs," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2047–2061, 2018.
[24] P. Sivakumar, M. NandhaKumar, R. Jayaraj and A. Kumaran, "Securing data and reducing the time traffic using AES encryption with dual cloud," in *IEEE Int. Conf. on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, 2019.
[25] Y. Li, "User privacy protection technology of tennis match live broadcast from media cloud platform based on AES encryption algorithm," in *IEEE 3rd Int. Conf. on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, pp. 267–269, 2020.
[26] T. Hidayat, S. Franky and R. Mahardiko, "Forecast analysis of research chance on AES algorithm to encrypt during data transmission on cloud computing," in *2nd Int. Conf. on Broadband Communications, Wireless Sensors and Powering (BCWSP)*, Yogyakarta, Indonesia, 2020.