Data security using cryptography with Image and Text De-duplication in Cloud

[1]Nayan Panpatil, [2]Madhavi Birla, [3]Pragati Aher, [4]Suvarna Kandekar

Department of IT(BE)
MET BKC, Nashik

Abstract: An effective storage and management of file systems is very much essential now a days to avoid the wastage of storage space provided by the cloud providers. Data deduplication technique has been used widely which allows only to store a single copy of a file and thus avoids duplication of file in the cloud storage servers. It helps to reduce the amount of storage space and save bandwidth of cloud service and thus in high cost savings for the cloud service subscribers. Today data that the proposed system need to store are in encrypted format to ensure the security. So data encryption by data owners with their own keys makes the de-duplication impossible for the cloud service subscriber as the data encryption with a key converts data into an unidentifiable format called cipher text thus encrypting, even the same data, with different keys may result in different cipher texts.

Keywords: Cloud computing, Data security, encryption and decryption, data storage in cloud.

Introduction: Cloud computing is network based computing system and it is the large storage space area where the authorized user can access the platform from anywhere and anytime with the good internet or network connectivity. Cloud computing is mainly to shared resources, hardware, software applications to provide the device with on demand. It is like a remote server on the internet to store, manage, and process data instead of using desktop. So, the working period is faster when compared to other local computers. Cloud computing is the information technology services and product. It support the virtualized resources, which is based on the reusability of IT infrastructure. Cloud Computing is the sequence of all required hardware, software, platform, applications, infrastructure and storage only with online identification. A proper security technique can make it safer and prevent data loses or stolen by hackers or intruders. Cryptography can make sure more security in information technology. A suitable encryption and decryption method can ensure data security in cloud computing. Various algorithm exists for cryptography such as DES, AES, and RSA etc.

Literature Survey:

In literature survey we learn guides or helps the researcher to define/find out/identify a problem. It is something when you look at a literature in a surface level, or an Ariel view. It includes the survey of place people and publications is context of research. [1] Medical organizations find it challenging to adopt cloud-based electronic medical records services, due to the risk of data breaches and the resulting compromise of patient data. Existing authorization models follow a patient centric approach for EHR management where the responsibility of authorizing data access is handled at the patients' end. This however creates a significant overhead for the patient who has to authorize every access of their health record. This is not practical given the multiple personnel involved in providing care and that at times the patient may not be in a state to provide this authorization. Hence there is a need of developing a proper authorization delegation mechanism for safe, secure and easy cloud-based EHR management. We have developed a novel, centralized, attribute based authorization mechanism that uses Attribute Based Encryption (ABE) and allows for delegated secure access of patient records. This mechanism transfers the service management overhead from the patient to the medical organization and allows easy delegation of cloud-based EHR's access authority to the medical providers. In this paper, we describe this novel ABE approach as well as the prototype system that we have created to illustrate it. [2] The rapidly growing demand for cloud services in the current business practice has favoured the success of the hybrid clouds and the advent of cloud federation. The available literature of this topic has focused on middleware abstraction to interoperate heterogeneous cloud platforms and orchestrate different management and business models. However, cloud federation implies serious security and privacy issues with respect to data sovereignty when data is outsourced across different judicial and legal systems. This column describes a solution that applies encryption to protect data sovereignty in federated clouds rather than restricting the elasticity and migration of data across federated clouds. [3] For building a secure cloud storage service on top of a public cloud infrastructure, attribute-based encryption (ABE) has been a preferred solution due to its flexible access control. ABE, however, incurs heavy computation cost on users during decryption. Thus, previous studies solved this problem by enabling cloud servers to perform a part of decryption operations on behalf of the users. In order to empower users to verify the correctness of the delegated decryption by the cloud, they employed a cryptographic commitment or message authentication code (MAC) to enable users to check the correctness of partial decryption of the cloud. However, the previous schemes fail to ensure the correctness of computation in the presence of malicious cloud servers. In this paper, we propose a novel and generic commitment scheme for ABE, which is secure against tampering attacks by malicious cloud servers. According to the performance analysis, the proposed scheme is only 0.5 ms slower on average than the previous commitment-based schemes and two to three times faster than the MAC-based scheme. [4] Storage-asa-service is an essential component of the cloud computing infrastructure. Database outsourcing is a typical use scenario of the cloud storage services, wherein data encryption is a good approach enabling the data owner to retain its control over the outsourced data. Searchable encryption is a cryptographic primitive allowing for private keyword based search over the encrypted database. The setting of enterprise outsourcing database to the cloud requires multi-user searchable encryption, whereas virtually all existing schemes consider the single-user setting. To bridge this gap, the system propose a practical multi-user searchable encryption scheme, which has a number of advantages over the known approaches. [5] Cloud computing is a new technology that transfer the computing process from personal computers into cloud servers over the internet. Nevertheless, as the client information data is stored in the cloud provider servers, the confidentiality of the information become a new concern. Different algorithms based on Encryption is presented previously to provide cloud clients with confidentiality. The main idea of encryption algorithms for cloud data is to permit cloud clients queries to be handled using encrypted data without decryption. This paper presents a new security mechanism using hybrid method of encryption algorithms and a distribution system to enhance cloud database confidentiality. A vertical fragmentation technique is adopted from alsirhani's model for distributing data over clouds. However, to overcome a weakness in alsirhani's model where compromises to a fragment can still make data meaningful. Instead, the proposed model uses a hybrid fragmentation technique to make data on fragments meaningless if compromised. The proposed model distributes the cloud database among the clouds using the provider views and level of confidentiality that is delivered by the employed encryption algorithms. To evaluate the proposed searchable encryption and hybrid fragmentation model, the study developed a Java application for simulating the hybrid cloud. The simulation combines public and private clouds; as essential processes is conducted inside the private cloud. The evaluation of the work was conducted by comparing the proposed model with existing solutions in query response and security characteristics. Preliminary results showed that the proposed searchable encryption and hybrid fragmentation model provides a secure mechanism that enhances data confidentiality in terms of faster response and additional security.

System Architecture:

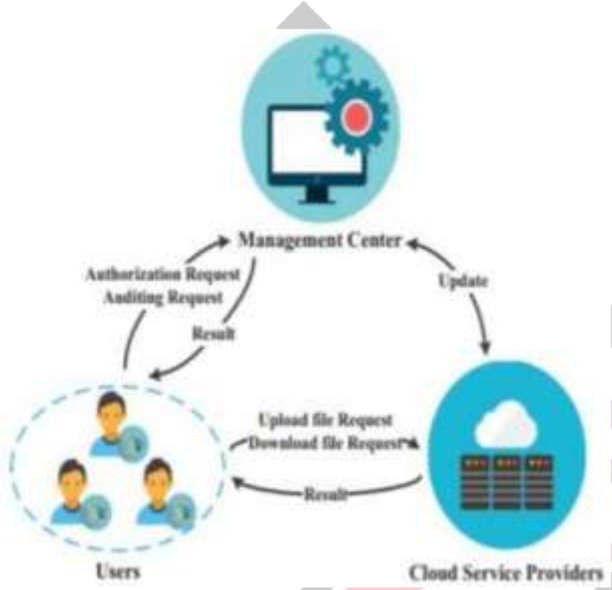


Fig.4. System Model of Secure Role Re-encryption System

A user sends a request to Management centre and encrypts the file, then it results as the cipher text to CSP. Users who belong to completely different role groups owning the corresponding role keys, with the role keys the user access cloud server, the user will upload or transfer the files from Cloud Service Provider. And user can download the file from the cloud server. A Cloud Service Provider is mainly for data storage, management and verification.

Cloud Service Provider stores and manages the uploaded files from authorized user Management centre is the trusted third party that is for the authorized user and for the role key management.

A. System Model: A user sends a request to Management centre and encrypts the file, then it results as the cipher text to CSP. Users who belong to completely different role groups owning the corresponding role keys, with the role keys the user access cloud server, the user will upload or transfer the files from Cloud Service Provider. And user can download the file from the cloud server. A Cloud Service Provider is mainly for data storage, management and verification. Cloud Service Provider stores and manages the uploaded files from authorized user Management centre is the trusted third party that is for the authorized user and for the role key management.

B. Adversary Model: A1 may has the communications between CSP and also the user to induce the transmitted info, and plays a job of the user to act with the CSP. A2 may listen the communications between CSP and also the user to urge the transmitted info, and cloud exchange S min bytes info with the user. A3 might discard the user’s knowledge that haven't been accessed or rarely accessed, and should tamper the user’s knowledge to take care of reputation. a role authorized tree to manage the user’s role and implement the role re-encryption key updating and revoking with efficiency, that satisfies the change of authorized user’s privilege. Once performing the secure knowledge deduplication, CSP will check the ownership of the licensed user. Security analysis shows that our proposed system is secure beneath the proposed security model, and performance analysis demonstrates the effectiveness and efficiency of our proposed system.

C. Design Goal: The main goal of the system is to protect the data from the cloud storage and the cloud server should be secure from the unauthorized user. So, then there will not any leakage in the cloud server.

D. Convergent Encryption: The same keys are always obtaining the same cipher text that can be by the two users with the new plaintext without the encryption keys.

E. Role Key Update: In role key updating, the generation of role keys is done in the management centre. The user can perform updating, downloading using the role re-encryption. And the management centre supports the role authorized tree.

Algorithm and methods:

I]Cryptography: is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process A new key (KEY1) using a hash function with stored KEY and delivers to the user/client by a secured channel (email, mobile etc.). User will enter KEY1 into the system. System will match this KEY1 with its previously generated KEY1. If

it. TYPES:1. Symmetric-key cryptography password. Secondly, verifying KEY1 (generated

successfully matched, system will generate again a KEY using anti-hash function from KEY1 and matched with its stored KEY. If matched again successfully, then system will treat this user as a valid user. This authentication system works in three steps. Firstly, supplying user ID and

2. Hash functions.3. Public-key cryptography

Our main goals are to secure stored data and authentication system in cloud environment. Many researchers tried to secure various credentials of users such as secure login, storing data/file with encryption, key management etc. By any means, if a hacker enters into the system, he may steal data/files from could end. If one intruder may successfully enter to cloud environment, then there is no way to detect him as a thief. By his credential he may access all data of the system. The system has offered such a system/environment that if any person enters into cloud end by any means, he will not succeed to get data/files from the cloud end. He will be caught. The procedures are described in next section.

A. Users Authentication:

When any clients/users will access his data/files or send new files, he has to login with his credential (user ID and password). If his credential is valid, then he will enter into next step of authentication. By this time, system will generate

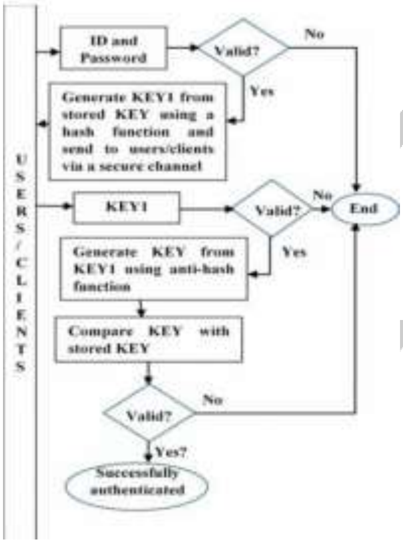

USERS/CLIENTS
ID and Password
Valid?
No
Yes
Generate KEY1 from stored KEY using a hash function and send to users/clients via a secure channel
KEY1
Valid?
No
End
Yes
Generate KEY from KEY1 using anti-hash function
Compare KEY with stored KEY
Valid?
No
Yes?
Successfully authenticated

Fig. 01 User authentication process

using a hash function with stored KEY). This KEY1 will be sent by a secured channel to the user. Thirdly, verifying stored KEY with newly generated KEY using anti-hash function with KEY1 (user supplied KEY1). As the system has stated earlier that, hackers will be caught even he supplies valid user ID and password. He will also be caught even while accessing the file that stored KEYs. As system will generate KEY1 using a hash function with stored KEY and sends KEY1 to the user by a secure channel, the invalid users/ hackers will not be able to access this KEY1. So, he will not be able to supply new KEY1. So there is no way to access data/file by an invalid user. An extra protection is also available by verifying stored KEY with newly generated KEY using anti-hash function with supplied KEY1. This system is described in Fig. 1 and Algorithm 1. After successfully login, user will able to access his data/file or send new file to cloud. Encrypted files/data will be decrypted using valid KEY and will send to the users.

B. Cloud End Auto Encryption: Auto encryption procedure is described in Fig. 2 and Algorithm 2. In the Fig. 2, the system has proposed an automated encryption system. This encryption system may be used hybrid cryptography system including RSA and AES or any other suitable encryption method. After login (described in previous section) users may access or send new data/files to store in the cloud and later he may logout. After successful logout, the system will lock those files/data, the user has been accessed or stored. Then system will create a new key (KEY2) by using a hash function with previously used KEY. Using this new key (KEY2), those files/data will be encrypted and new key(KEY2) will be replaced with previous stored KEY.

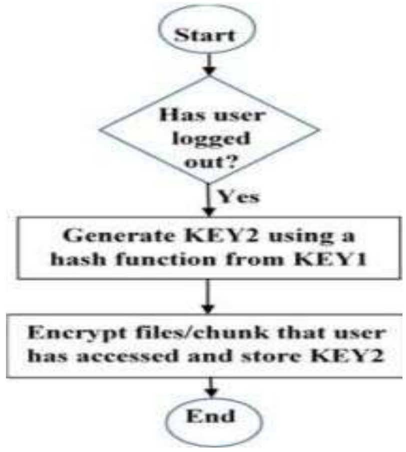

Start
Has user logged out?
Yes
Generate KEY2 using a hash function from KEY1
Encrypt files/chunk that user has accessed and store KEY2
End

Fig. 02 Cloud end auto encryption

C. Algorithm 1:

1. User login will be verified by a suitable system. 2. If login credential is valid, generate KEY1 using a hash function with stored KEY and send to the user by a trusted/secured medium. 3. Provide new KEY1 (by the user) to get a chunk of encrypted data/files. 4. Compare provided KEY1 with system's KEY1, if matched, then go next step, otherwise exit. 5. Generate KEY using anti-hash function with user supplied KEY1; match this KEY with stored KEY; if does not match, then exit. 6. Decrypt data chunk by this KEY and send to the user. 7. If user wants to store file, then encrypt it by suitable encryption algorithm. D. Algorithm 2:

1. At first, User login will be verified Algorithm 1. 2. If user logout, then generate KEY2 using a hash function with KEY1. 3. Encrypt files/data that has been accessed/stored by this user using a suitable encryption algorithm. 4. replace KEY1 by KEY2 and exit.

II] Image upload process:

For image uploading the user has to click the image upload button at the top of the Cloud system platform. Then from the user's image storage he/she has to select any image with any size and with any dimension. If some other user uploads the same image with same size, dimension then it will not upload into the cloud system (i.e. it gives notification as `the image is already present' to the user) And if the same user try to upload the same image with different size, dimension then it automatically uploaded into cloud system directly

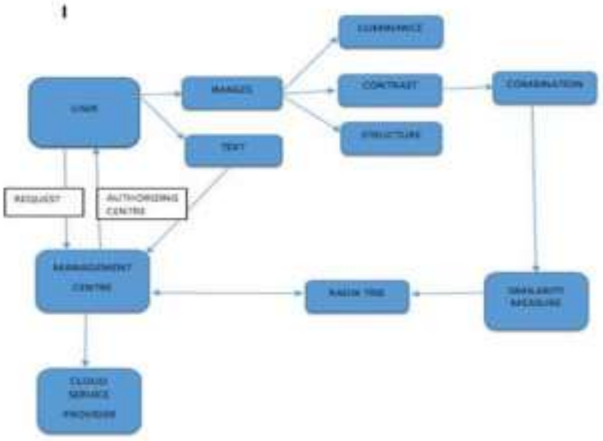


Fig.3. Image and text avoidance in cloud server

For secure Image Upload Process proposed system is using Radix trie method. The proposed system need to avoid the similar image i.e. For this the proposed system will use the Image pixel value. The pixel value of the image is calculated by Python Imaging Library. If the authorized user uploads the image of some pixel dimension, then some other authorized user uploads the same image with slight change then automatically it occurs the radix trie operation. For example, if the first image pixel value is 256 and the pixel value of another image is 255. Then from the first image pixel value the second image pixel value is derived. As Radix Trie generates encoded names and passes to the Radix Trie structure, these encoded names are lookup in the Radix Trie. If the encoded name does not exist already, an insert operation is performed for an encoding name. For a request, update , insert and delete operations are done on the Radix Trie.

If the pixel values are (165 , 168 , 167 , 142 , 149) then the radix tree is looks like as shown in fig.4,

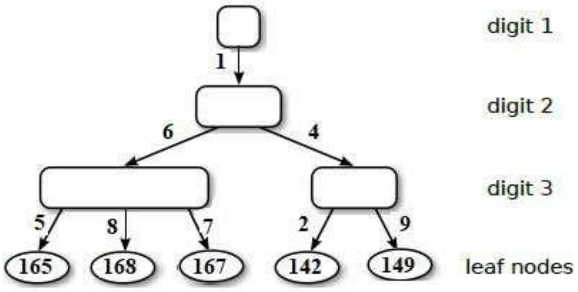
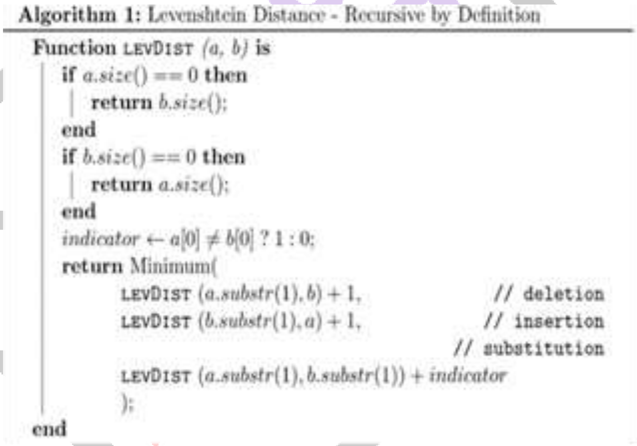


Fig.4. An example of radix trie method

III] Text upload process:

If the user wants to upload any text file then user has to click the text upload button at the top of the cloud system platform, Then the text file which user has to be ready in their normal server storage in their platform, after this user has to click the choose file button then it goes to that saved area and it automatically inserted and user has to click the upload button, then it automatically uploaded. If some other authorized user uploads the same file then that particular file will not be uploaded and also it exists. For the text upload process. The proposed is using 3 main algorithms,
1. Levenshtein string distance algorithm 2.Fuzzy string matching algorithm 3.Dice coefficient algorithm

1] Levenshtein distance algorithm: The Levenshtein distance is a string metric for measuring difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits i.e. insertions, deletions or substitutions required to change one word into the other. **Levenshtein distance is the smallest number of edit operations required to transform one string into another.** Edit operations include insertions, deletions, and substitutions. Levenshtein algorithm is also known as edit distance algorithm in Metrix.



Algorithm 1: Levenshtein Distance - Recursive by Definition

```
Function LevDist (a, b) is
    if a.size() == 0 then
        return b.size();
    end
    if b.size() == 0 then
        return a.size();
    end
    indicator ← a[0] ≠ b[0] ? 1 : 0;
    return Minimum(
            LevDist (a.substr(1), b) + 1,                // deletion
            LevDist (b.substr(1), a) + 1,               // insertion
                                                      // substitution
            LevDist (a.substr(1), b.substr(1)) + indicator
            );
end
```

fig. Levenshtein Distance Algorithm Hash algorithm :

A hashing algorithm is a mathematical algorithm that converts an input data array of a certain type and arbitrary length to an output bit string of a fixed length. Hashing algorithms take any input and convert it to a uniform message by using a hashing

Hashing Algorithm
Plain Text
#
Hash Function
Hashed Text
cheapSSL

2] Fuzzy string matching algorithm: Fuzzy Matching also called as Approximate String Matching is a technique that helps identify two elements of text, strings, or entries that are approximately similar but are not exactly the same. Fuzzy logic is a form of multi-valued logic that deals with reasoning that is approximate rather than fixed and exact. Fuzzy logic values range between 1 and 0. i.e. the value may range from completely true to completely false.

Architecture of Fuzzy Logic:

In the architecture of the **Fuzzy Logic** system, each component plays an important role. The architecture consists of the different four components which are given below.
1. Rule base 2.Fuzzification3.Inference engine 4.Defuzzification

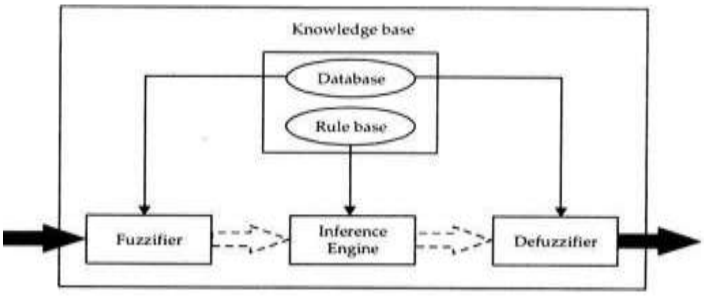


Fig. Architecture of Fuzzy logic.

1. A fuzzifier, which translates crisp (real-valued) inputs into fuzzy values.2. An inference engine, which applies a fuzzy reasoning mechanism to obtain a fuzzy output using the rules contained in the knowledge base. These fuzzy rules, which define the connection between input and output fuzzy variables have: IF antecedent THEN consequent format.3. A de-fuzzifier, which translates this latter output into a crisp value;4. A knowledge base, which contains both an ensemble of fuzzy rules, known as the rule base, and an ensemble of membership functions known as the database.

3] Dice coefficient algorithm:

The dice coefficient measures how similar a set and another set are. It can be used to measure how similar the two strings are in terms of the number. Basically it uses union operation which measures the similarity of two strings besides their position

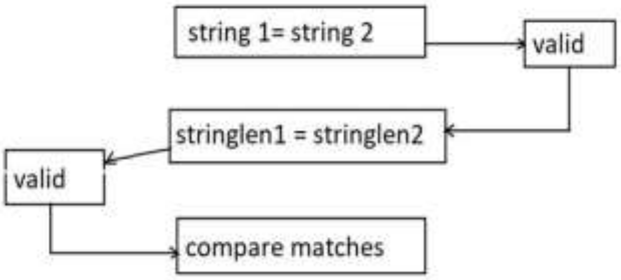


Fig. Dice coefficient logic

Proposed Work:

The proposed system is creating a system that allows user to provide security to their files/data and protect them from hackers and avoid the malicious attacks. The proposed system is using hash function for encryption of the password and yet by chance if the hacker breaks the hash code; he will get only the dummy data by the system. OTP(one time password) will also be there for authentication using user's email or mobile number.

Network traffic in the Cloud encryption environment is characterized by large scale, high dimensionality, and high redundancy, these characteristics pose serious challenges to the development of cloud. An effective stacked contractive auto encoder (SCAE) method is presented for unsupervised feature extraction

References:

[1] Maithili Arjunwadakr and R.V. Kulkarni, "The rule-based Intrusion Detection and Prevention Model for Biometric System", Journal of Emerging Trends in Computing And Information Sciences, OCT 2010.

[2] Todd Vollmer, Jim Alves-Foss and Milos Manic, "Autonomous rule creation for intrusion detection", 2011 IEEE Symposium on Computational Intelligence in Cyber Security.

[3] M. Sebring, E. Shellhouse, M. Hanna and R. Whitehurst, "Expert systems in intrusion detection: A case study", Proceedings of the 11 th National Computer Security Conference, pp. 74-81, 1988.

[4] T. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. Edwards, P. Neumann, et al., IDES: The Enhanced Prototype. A Real-Time Intrusion Detection System, 1988.

[5] D. Anderson, T. Lunt, H. Javitz, A. Tamaru and A. Valdes, Detecting Unusual Program Behavior Using the Statistical Component of the NextGeneration Intrusion Detection Expert System (NIDES).

[6] H. Anggeriana, S. Kom and M. Kom, "Cloud Computing", Jurnal Teknik Informatika, vol. 1, 2011.

[7] T. Velte, A. Velte and R. Elsenpeter, Cloud computing a practical approach, McGraw-Hill, Inc, 2009.

[8] P. Mell and T. Grance, "The NIST definition of cloud computing", National Institute of Standards and Technology, vol. 53, pp. 50, 2009.

[9] A. Yousif, M. Farouk and M. B. Bashir, "A Cloud Based Framework for Platform as a Service", in Cloud Computing (ICCC) 2015 International Conference on, pp. 1-5, 2015.

[10] E. Hossny, S. Khattab, F. Omara and H. Hassan, "A Case Study for Deploying Applications on Heterogeneous PaaS Platforms", in Cloud Computing and Big Data (CloudCom-Asia) 2013 International Conference on, pp. 246-253, 2013.

[11] M. O. Imam, A. Yousif and M. B. Bashir, "A Proposed Software as a Service (SaaS) Toolkit for Cloud Multi-Tenancy", Computer Engineering and Applications Journal, vol. 5, 2016.

[12] M. M. Alani, Elements of cloud computing security: A survey of key practicalities, Springer, 2016.

[13] A. team, Open Source Metadata-Based Java ORM Framework for Cloud SaaS Applications, 2011,
[online] Available: http://www.athenasource.org/java/

[14] S. Paliwal, "Cloud application services (SaaS)-Multi-Tenant Data Architecture", Infosys technologies limited, Sep 2014, [online] Available: http://www. cmg. org/wpcontent/uploads/2012/11/m_94_4. pdf.

[15] S. A. Elmubarak, A. Yousif and M. B. Bashir, Performance based Ranking Model for Cloud SaaS Services, 2017