Article

Big Data Handling Approach for Unauthorized Access of Cloud Computing

Abdul Razaque [1]*, Shaldanbayeva Nazerke [1],Bandar Alotaibi [2,3]*, Munif Alotaibi [4]*, Akhmetov Murat [5], Aziz Alotaibi [6]

[1] Department of Cybersecurity, IITU, Almaty 050000, Kazakhstan; a.razaque@iitu.edu.kz (A.R.); 24795@iitu.edu.kz (B.V.); s.amanzholova@iitu.edu.kz (S.A)
[2] Sensor Networks and Cellular Systems Research Center, University of Tabuk, Tabuk 71491 , Saudi Arabia; b-alotaibi@ut.edu.sa
[3] Department of Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
[4] Department of Computer Science, Shaqra University, Shaqra 11961, Saudi Arabia; munif@su.edu.sa
[5] Department of Information Security, L. N. Gumilyov Eurasian National University, Kazakhstan
[6] Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944. Saudi Arabia , Saudi Arabia; azotaibi@tu.edu.sa
* Correspondence: a.razaque@iitu.edu.kz, b-alotaibi@ut.edu.sa, munif@su.edu.sa

Abstract: Nowadays, cloud computing is one of the important and rapidly growing paradigms that extend its capabilities and applications in various areas of life. The cloud computing system challenges many security issues, such as scalability, integrity, confidentiality, and unauthorized access, etc. An illegitimate intruder may gain access to the sensitive cloud computing system and use the data for inappropriate purposes that may lead to losses in business or system damage. This paper proposes a hybrid unauthorized data handling (HUDH) scheme for Big data in cloud computing. The HUDU aims to restrict illegitimate users from accessing the cloud and data security provision. The proposed HUDH consists of three steps: data encryption, data access, and intrusion detection. HUDH involves three algorithms; Advanced Encryption Standards (AES) for encryption, Attribute-Based Access Control (ABAC) for data access control, and Hybrid Intrusion Detection (HID) for unauthorized access detection. The proposed scheme is implemented using Python and Java language. Testing results demonstrate that the HUDH can delegate computation overhead to powerful cloud servers. User confidentiality, access privilege, and user secret key accountability can be attained with more than 97% high accuracy.

Keywords: Data security; data handling; access control; unauthorized access; cloud computing

Citation: Razaque, A.; Nazerke, S.; Alotaibi, B.; Alotaibi, M.; Murat, A.; Alotaibi, A. Big Data Handling Approach for Unauthorized Access of Cloud Computing. *Preprints* **2021**, *1*, 1. https://doi.org/

Academic Editor:

Received: 9 October 2021
Accepted: 2 November 2021
Published:

1. Introduction

Cloud computing is one of emerging paradigms which is popular nowadays. Cloud computing companies provide almost all possible ways to process data: storing it, changing it, sharing it with anyone else, and eventually deleting it [1], [2]. But the most attractive feature, that distinguishes them from other traditional storage systems, is their convenience: data can be obtained anywhere and anytime instantly if you have a connection to the Internet [3]. Nowadays people are looking for convenient, fast, inexpensive systems to facilitate their tasks. This has become a factor in creation of the cloud systems, which responds to these characteristics mostly and also has a serious problem with the security of data that was provided by users [4].

Depending on what service organizations and individuals need, cloud computing can be characterized by three existing models, namely software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) [5], [6], [7]. SaaS is some software applications that were provided to the customers so that users can use it in the cloud without any installation on their desktops, laptops and so on. Today there exist various platforms around the world that you can use [8]. The popular examples are as follows: Amazon's EC2 [9], Amazon's S3 [10], IBM's Blue Cloud [11], Google App Engine [12], Yahoo Pig, Google's Apps [13], Dropbox [14], and Salesforce's Customer Relation Management (CRM) System [15]. PaaS is a platform for developing software applications provided by cloud

 doi:10.20944/preprints202112.0068.v1

computing e.g., Amazon Web Services [9], Window Azure [16]. IaaS is infrastructures, like servers, operating systems, networks and so on, provided for users through virtualization. virtualization is principal enabling core of cloud computing. Virtualization uses software to split one computer device into multiple independent computing devices where each one of them can be used to perform computing tasks. It help us to efficiently allocate and utilize the usual idle computing resources, reduces cost, and reliably increase infrastructure utilization. Among these are DigitalOcean, Linode, Rackspace, Microsoft Azure, GCE and so forth [17], [18], [19]. These systems significantly increase work efficiency in organizations at a relatively low price.

These organizations provide services as a pay-as-you-use manner at a relatively low price. For companies, it is rather easier to use cloud systems, because users will save the investments that would have been paid for building their infrastructure [20].

As long as these given benefits, there are also other advantages, like increased efficiency, portability, scalability, flexibility and so on [21].

Cloud computing provides resources in computing infrastructure over the network for the organizations to use. With the existing crucial benefits, there are several challenges related to the security of access control and privilege managements [22].

Cloud computing provides resources in computing infrastructure over the network for the organizations to use.

Scalability, integrity, data access are examples of the security and privacy challenges that face every user in cloud computing . The data access is a significant service in the cloud without which this platform would not be able to operate and be so popular among the users. Therefore, unauthorized access in the cloud system is one of the important problems that must be solved or prevented to such an extent, when every user would be able to trust their sensitive data to the provider [23].

Thus, security and privacy of cloud data is important as people and organizations are very concerned that their data might fall into the hands of third parties who can use them for their purposes [24]. For example, cloud data need to be secured within a trusted domain, e.g. data owners, industry and big organizations data, and federal agents. Also, cloud data need to be saved in a fully trust cloud data base. Unauthorized data access is one of the existing security issues of the cloud computing system [25], which should be solved and improved continuously.

Several mechanisms have been proposed in the past for secure data access in cloud computing. These works considered various problems in cloud computing and tried to achieve fine-grained, scalable and secure data access control. For example, the authors of [17] combined three cryptographic techniques, namely KP-ABE, PRE, and lazy re-encryption. Each of them solves specific issues, making in a whole fine data access system. But when it comes to multiple-levels of attribute authorities the system cannot properly handle this. To solve this problem, [18] introduced a new approach that extend to [17] solution. To achieve flexible and scalable data access control in cloud computing, the authors of [17] implemented a hierarchical attribute-set-based encryption (HASBE) algorithm. In [19] combined two existing algorithms, namely hierarchical identity-based encryption (HIBE) and cipher-text policy attribute-based encryption (CP-ABE). After this [19] made a performance to expressivity tradeoff and then used proxy and lazy re-encryption algorithms on the given output. Other works [26–30] also proposed other solution for secure access control, which are described in detail in section 3. Nevertheless, those works still have their own disadvantages and shortcomings. For examples, these methods are still susceptible to several type of cyber threats. Some of them use public keys encryption which is very slow when compared to the symmetric encryption, and has many potential certification issues since it depends on third party. Different from the existing method, The HUDH will consist of data encryption, data access, and intrusion detection, will address the unauthorized access in cloud computing systems as described in the next sections.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

2. Problem Statement

Security has been always an essential issue for any computing environment [31], where the data, hardware, and software should be protected from unauthorized access. A could service provider in cloud computing offers computational services and virtualizations over the internet. It comprises some critical factors such as restricting unauthorized access, maintaining data integrity, and ensuring data availability. To assure security in cloud, heightened security challenges must be addressed to take full advantages of this computing paradigm [32]. To go deep into the origins of unauthorized access in cloud computing systems as to what causes it, a proper research methodology was followed. The existing schemes were studied in accordance with research strategies to address the problem of unauthorized access in cloud computing. Thus, an efficient model of access control, which includes encryption, data access, and intrusion detection algorithms, is required to secure data in the cloud as well as take measures against intrusion [33].

Cloud computing involves enormous number of devices, applications, and parties that make designing a secure data sharing framework a difficult task to accomplish. Moreover, data on the cloud are susceptible to various threats such as losses, accidentally altered by the cloud provider, and attacks [34]. Thus, developing a complete security method for cloud storage is inevitable.

Thus, we review the existing methods of cloud computing systems, we investigate the use of advanced security mechanisms such as advanced data encryption, secure data access, and accurate intrusion detection mechanism to build secure cloud computing model, we use poplar dataset to test and analyze the proposed model against many types of attacks.

2.1. Motivation

Building security mechanism for cloud storage is very essential task. legitimate participants who want to share their data on the cloud want secure controlling and accessing mechanisms, as well as fast and safe sharing of data on the demand. Moreover, the existing has several shortcomings. Thus, we need to has robust cloud computing system that can provide advanced data encryption, secure data access, and accurate intrusion detection mechanism.

2.2. Contributions and Organization

In this paper, a mechanism for handling unauthorized data access in cloud computing is proposed which consists of several steps, namely data encryption, data access, and intrusion detection [35–37]. To implement these mechanisms, we use some existing efficient algorithms that are Advanced Encryption Standard (AES) for encryption, Attribute-Based Access Control (ABAC) algorithm for data access control, and Hybrid Intrusion Detection (HID) algorithm for unauthorized access detection.

- HUDU scheme integrates three state-of-the-art algorithms (ABAC, AES, and HIDS) to ensure data security, user authentication, and prevention of potential threats in cloud computing.
- The HIDS algorithm combines the features of two known algorithms (Random Forest and neural network) for important feature selection and training the data. The higher accuracy is achieved With the integration of these two algorithms.
- The proposed HUDU is tested on the known UNSW-NB15 dataset using class-4 and class-6 to confirm the accuracy. Deployment of different classes provides a new direction. As higher accuracy is achieved when employing a higher class.

This paper is organized as follows. Section 3 reviews the existing schemes to solve unauthorized data access problem in cloud computing environments. Section 4 presents a detailed description of the proposed scheme (HUDH) which involves three algorithms; Advanced Encryption Standards (AES) for encryption, Attribute-Based Access Control (ABAC) for data access control, and Hybrid Intrusion Detection (HID) for unauthorized access detection. Section 5 shows the implementation and presented the results in details. Section 6 discusses the results. Section 7 concludes the paper.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

3. Related Works

In this section, we review the existing schemes and models which tries to solve unauthorized data access problem in cloud computing. All these works provided several methodologies for this issue, which we will describe in general and show the advantages and disadvantages of their solutions.

A Large-Universe Attribute-Based Encryption with Public Traceability for Cloud Storage is proposed [38]. This work mainly addresses both key abuse and key escrow concerns when deploying ABE in a cloud computing environment. Two different approaches are used, key generation center (KGC) and an attribute authority (AA), participate for the generation of the user's secret key. Both KGC and AA will not know the full decryption key or have the capability to forge one. However, such model is require expensive computations in the cloud.

To achieve secure, fine-grained and scalable data access on the cloud, [17] combined three leading cryptographic techniques, namely KP-ABE, PRE, lazy re-encryption and applied it to the data being stored in the cloud. A set of attribute are added to the data file and gave each user access structure to that set of attributes in the form of a tree. To make it possible KP-ABE algorithm is used to escort data encryption keys of data files. This facilitated the fine-grandness of access control. But this algorithm, if used alone, would put a lot of computational overhead to the owner, because users are charged for the utilization of the given services. Mostly this overhead comes from user revocation operation, where the owner should re-encrypt all the data files accessible to the user who is leaving the system, even it is required to the owner be online all the time while the revocation is running. To solve this problem a hybrid algorithm is introduced that included PRE and KP-ABE algorithms and made in the way where the owner can delegate the computational overhead to the Cloud Server itself. The Server from his side cannot read the files because of the attributes in the file. And this helped the owner to get rid of the extra overhead and be able to control the data in the cloud. But in this case, the server has the computational overhead. To reduce it, another algorithm known as Lazy re-encryption is added, which helped the Server to unite the tasks of some computational operations. The operation on the server is independent of the number of the users because the computations are done on the attributes in the data file or on the size of the access structure in the form of the tree. It means it does not matter how many users will be added or removed, the data access will not be corrupted. Therefore the algorithm can provide scalability. But with these advantages, there is also a problem of scalability when it comes to multiple-levels of attribute authorities.

A novel digital forensic architecture using fast-growing Software-Defined Networking (SDN) and Blockchain technology for Infrastructure-as-a-Service (IaaS) cloud [39]. Secure Ring Verification based Authentication (SRVA) scheme is proposed to protect the system from unauthorized users. To strengthen the cloud environment, secret keys are generated using Harmony Search Optimization (HSO) algorithm. For encryption, Sensitivity Aware Deep Elliptic Curve Cryptography (SA-DECC) algorithm is used.

A sensitive and energetic access control (SE-AC) mechanism is proposed for providing a secure access control even in critical situations [40]. The proposed mechanism ensures the confidentiality of data, by authorizing individuals to have limited permission to edit or modify patient's data. Data is encrypted before it is send to the cloud storage. The user can get require access his permissions are changes based on authentication and context attributes. In addition, the access operation to IoT could are controlled and assessed for security analysis to prevent any unauthorized access.

A secret sharing group key management protocol (SSGK) to protect the communication process and shared data from unauthorized access is proposed [10.4]. Different from the prior works, a group key is used to encrypt the shared data and a secret sharing scheme is used to distribute the group key in SSGK. The extensive security and performance analyses indicate that our protocol highly minimizes the security and privacy risks of sharing data in cloud storage and saves about 12% of storage space.

A novel fog-centric secure cloud storage scheme to protect data against unauthorized access, modification, and destruction is presented in [34]. To prevent illegitimate access, the proposed scheme employs a new technique known as Xor-Combination to conceal data. Moreover, Block-Management outsources the outcomes of Xor-Combination to prevent malicious retrieval and ensure better recoverability in case of data loss. The proposed approach is based on hash algorithm for the detection of malicious attacks with higher sensitivity. The robustness of the scheme is shown through security analysis and experimental results to validate performance of the proposed scheme in terms of data processing and time.

The work of [18] proposed Hierarchical Attribute-Based Encryption (HASBE) algorithm, which is the extension of the Attribute-Based Encryption algorithm with added hierarchical structure delegation algorithm which is close to CP-ABE scheme. In addition, the security of the HASBE is proved, because the CP-ABE algorithm which is similar to their hierarchical structure is secure under some models of vulnerabilities. The scalability is also achieved, which is done by extending the ASBE with the hierarchical structure. It is used to delegate the generation operation of the user's private attribute to the lower-level domains. As compared to [17] these works provide flexibility by making the organization of the user attributes recursive structure, which makes it possible for the user to impose dynamic constraints on those attributes combinations to satisfy the policy. So the given algorithm can handle complex attributes and multiple operations for given attributes. Because the HASBE is based on ASBE the algorithm can provide fine-grained access to the users, as well as efficient user revocation by giving the attributes to each user's key and provide multiple value assignments to those attributes. Therefore, this algorithm suits well to solve the given issue.

[19] proposed algorithm which helps companies to share data on cloud servers efficiently. This solution combines two existing algorithms, namely hierarchical identity-based encryption (HIBE) and cipher-text policy attribute-based encryption (CP-ABE). The proposed method made a performance to expressivity tradeoff and then used proxy and lazy re-encryption algorithms on the given output. This work has several advantages, like high performance, fine-grained access control to the system, scalability and full delegation. The HABE scheme is known as collision-resistant, which was already proved to be secure against random and adaptive attacks. This model includes root master (RM) which has its respect third trusted party (TTP), domain masters (DM) and the users which are the personnel of the organization. The RM is used for the generation of the parameters and keys of the domain and the distributions of these results, while DM is used to delegate the keys to the next level and the users. In the HABE model, each DM and attribute with an ID are marked while each user was assigned with ID as well as the attribute. After this [19] made the object's private key to be extracted from DM and the public key is the combination of the public key and the ID of the DM. This algorithm shows how data can be efficiently delegated and shared in the cloud server.

[26] proposed a novel framework for access control to PHRs within the cloud computing environment. To enable fine-grained and scalable access control for PHRs, leveraged attribute-based encryption (ABE) techniques to encrypt each patient's PHR data. To reduce the key distribution complexity, divided the system into multiple security domains, where each domain manages only a subset of the users. By doing so every user, i.e. the patient can be able to fully control the privacy of the given data, and also it reduces the complexity of the key management in the system. The proposed solution gives the system flexibility, efficiency in user revocation and data access in emergencies. Advantages of their methodology: greatly facilitates the storage, PHR service providers to shift their PHR applications and storage into the cloud, to enjoy the elastic resources and reduce the operational cost. As well as this, the work has a number of disadvantages: by storing PHRs in the cloud, the patients lose physical control to their health data, which makes it necessary for each patient to encrypt her PHR data before uploading to the cloud servers; the key distribution can be inconvenient when there are multiple owners, since it requires each owner to always

be online; also in this paper discussed methods for enabling efficient and on-demand revocation of users or attributes, and break-glass access under emergency scenarios. It is leveraged attribute-based encryption (ABE) techniques to encrypt each patient's' PHR data. Also, there was used Public key cryptography (PKC) and symmetric key cryptography (SKC) based solutions.

[27] presented a framework, known as EUCALYPTUS which is based on the Infrastructure as a Service (IaaS) model. This system provides the owner with virtual machine instances that are distributed in different physical resources. In the work precisely described the EUCALYPTUS algorithm, the system's operational aspects and architectural structure, which directly affect the scheme, like the algorithm's portability, modularity, and simplicity. Also, there is evidence which states that EUCALYPTUS encourages users, who used systems like Grid and HPC, to explore other functionalities of the system while working with it. Implemented each component as a stand-alone Web service, which can help to make the system modular. This also provides the system with the following benefits: a well-defined API in the form of WSDL, which contains operations that the system can perform, and input/output data structures. Also, users can use those existing features of the service for secure communications among components.

[28] reviewed various features of an attribute-based access control mechanism, suitable for a cloud computing environment. It led to the design of an attribute-based access control mechanism for cloud computing. Access control decisions are important for any shared system. But for cloud computing, the factors like scalability and flexibility is crucial. Various access control methods are used in cloud computing and highlighted features of attribute-based access control, which are important for designing an attribute-based access control. Many access control methods have been considered and also emphasized their benefits and weaknesses. In identity-based access control models including Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role-Based Access Control (RBAC) user (subjects) and resources (objects) are identified by unique names. The benefit of using them is that these access control methods are effective in the unchangeable distributed system, where there are only a set of Users with a known set of services. Nevertheless, in large distributed open systems, users and resource providers are not in the same security domain. The users are characterized by their attributes and properties rather than by predefined identities. So, identity-based access control models are not much effective in these systems. This is the main weakness of them. RBAC provides access to the user based on the job role. In RBAC permissions can be added or deleted if the privileges for a role change, so the main advantage is dynamically changeable permissions. But it's difficult to reach an agreement on what privileges to associate with a role. In Attribute-based access control (ABAC) access is granted on attributes that the user could prove to have such as date of birth or national number. Attribute-based access control extends role-based access control, in general, with the following features: delegation of attribute authority, decentralization of attributes and interference of attributes. It is popular because it is flexible to support various kinds of domains and policies. The disadvantage is due to several factors: reaching an agreement on a set of attributes is hard, especially across multiple agencies or domains and organizations. The existing methods of access are based on the authentication of the user from one site, as well as from the time the request is done. Sometimes it is labeled as authentication based access control. It is required to tightly couple domains in all of these methods. This is made to combine identities or to identify the meaning of attributes and roles. Also, this approach makes hard to assign a set of administrator attributes to users. [29] discussed various features of the attribute-based access control mechanism, suitable for a cloud computing environment. It led to the design of an attribute-based access control mechanism for cloud computing. ABAC provides policies for the sensitivity of the important data of the user. But as well as this, it provides autonomy to the organization while working with it. Also it automated trust negotiation, which can be checked when it is required. Each object in the system has attributes that defined the identity and characteristics of its respective object. These policies are supported by the authorization system of the server. Each system has its

policy description methods. To make policies able to integrate the system and to provide scalability to the data access, each of these rules is encapsulated as an independent unit. This work was important for designing an attribute-based access control.

[30] introduced a technique based on the capability to solve data access control which enables only valid users can have access to the data in the cloud. Also, modified the Diffie-Hellman protocol for key exchange between the user and provider to secretly share each other symmetric keys. This provided secure data access D-H key exchange model to cloud system. The results of their analysis showed that the technique is quite efficient and secure on some existing security models. Also, it includes some set of security protocols that handles the data access control in the cloud infrastructure. Furthermore, the proposed solution includes a capability-based model as well as public-key encryption to protect sensitive data in the cloud. For the users provided D-H key exchange model to securely access data from another domain. This solution does not have the user's public key, making the key management of the system easier. The ciphers of the public key, private key and of the hash are located in the isolated and secure place in the cloud. The proposed approach also encourages the owners to store their sensitive data in the cloud without loss of control to it. It also delegates the computational overhead to the server, making it more attractive to users.

The authors of [41] introduced a secure cloud-based IoT data management model to efficiently manage cloud storage and bandwidth. The proposed model is also capable of tracking traitors who illegitimately reveal their secret keys to other parties. The proposed method consists of three phases: (1) each IoT user has a specific transformation key; the key holder identity is authenticated in the cloud, (2) the ciphertext is partially decrypted utilizing the transformation key of the IoT user, (3) eventually, the partially decrypted result is returned. Although, this approach can securely and efficiently manage cloud storage, however, it is required to install a cloud server to perform decryption and to reduce the computation overhead associated with decryption.

In cloud computing, both blockchain and deep learning has shown to be successful in many cases particularly to improve the security of the cloud. For example, the authors of [42] has used both blockchain and deep learning algorithm named a bidirectional long short-term memory (BiLSTM) to improve data privacy and detect cyberattacks, Moreover, the deep learning algorithm is used to detect and identify any malicious malware or virus or while the blockchain is used to provide more robust data privacy. However, BiLSTM, which has double LSTM cells, is known to be very heavily, costly and difficult to train.

Moreover, the authors of [43] used RNN model to improve the security of cloud services partuclry against cloud malware. cloud malware is software that widely used to attack VMs hosted on a cloud IaaS. The authors used two diffrent types of RNN algorithms which are Bidirectional recurrent neural networks (BRNN) and Long short-term memory (LSTM) to Detecting malware in a rapid and effective way. The model,is trained to detect the behavior of malware. It achieved high accuracy over 99%. However, both LSTM and BRNN are requires large dataset to be trained.

Simmlary, other deep learning techniques such as Convolutional Neural Networks (CNNs) [44,45] have used for clould security. Although, deep learning techniques have good and acceptable performance, but we need more light-weighted technique with minimum complexity that can be easily and suitably deployed in the cloud. Deep learning techniques have many layers, complex structures, and the huge number of neurons and parameters. Thus, they are computationally expansive, requires high computing power and large network capacity.

The author of [46] utilized three well-known techniques namely, attribute-based encryption (ABE), identity-based timed-release encryption (IDTRE), and distributed hash table (DHT) to design a secure and efficient cloud computing access control method. The aim of this access control is to provide cloud computing environment with resource and knowledge sharing. Initially, the author used users attributes to encrypt the data. Then, the encrypted data is split into two parts: the extracted ciphertext and the encapsulated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

ciphertext. Consequentially, the ciphertext shares are generated by encrypting the decryption key and combining both the extracted ciphertext and the key's ciphertext. Finally, the author suggested to distribute the ciphertext shares into the DHT network and store the encapsulated ciphertext on the cloud servers. The proposed approach proves its effectiveness compared to related work through security and performance analysis. However, combining more than encryption methods increases the complexity.

Namasudra et al. [47,48] proposed an access control method based on the popularity value to provide cloud computing users with efficient data accessing. The proposed approach relies on public key to provide requested data to users while decreasing the access time and searching cost. Moreover, the security of data confidentiality for users is maintained. A significant aspect of this proposed approach is its capability to resist several attacks such as phishing, stolen-verifier, masquerade, man-in-the middle, and internal attacks. However, public-key encryption is slower than symmetric encryption; thus it would be preferable if the author have used symmetric encryption because it is faster.

4. Proposed Hybrid Unauthorized Data Handling

We have designed a mechanism that considers data access control from a different point of view. One will protect the data if the intruder gets access to it by encrypting it. The other algorithm will handle the access issue, and will safely delegate the rights to the user. And the last algorithm will provide fast detection of the intrusion to the system, which makes it possible to react quickly.

When the data is loaded to the cloud, the system automatically applies the AES algorithm to encrypt it and store it in the database. Each data owner will have a secret key and the consumer of that data with the public key. Therefore, if the data is on the hand of the third party it won't be possible to use it, because there is no key to decipher it. Each user, data and actions will have their own attributes, which will be used by the ABAC algorithm to find relationships among them and give access if the policy allows it. The intrusion detection system will work when the session of the user starts and will monitor until the session ends. If the system identifies anomaly behaviors of the user it will instantly send the notification message to the administrator.

Algorithm 1 Encryption

Input: $\{D, I_k\}$ in
Output: $\{D_e\}$ out
1: **Initialization:** $\{$*D: data,* D_b*: database,* D_e*: encrypted data, A: AES algorithm, N: number of iterations,* I_k*: initial key,* R_k*: round key,* B_r*: byte rows,* B_c*: byte columns, B: bytes* $\}$
2: **Do**
3: **Add** R_k
4: **Apply** A on D by shifting B
5: **Shift** B_r
6: **Mix** B_c
7: **While** $N \cong 9$
8: **Get** D_e
9: **Store** $D_e \rightarrow D_b$

In algorithm 1, Step-1 shows the initialization process of used variables. The input and output processes are shown at the beginning of the algorithm. Step-2-7 apply the AES cryptographic algorithm to get encrypted data, by substituting bytes, shifting rows and mixing columns.

Step-8 shows created encrypted data. Step-9 shows the process of storing the encrypted data into the database.

In algorithm 2, Step-1 shows the initialization process of used variables. The input and output processes are shown at the beginning of the algorithm. Step-2 shows the process of defining policies in the JSON file. Step-3 makes request to JSON file to get access for data.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

Algorithm 2 Data Access

Input: $\{U, R_d\}$ in
Output: $\{A\}$ out
1: **Initialization:** $\{D_b$: *database*, D_e: *encrypted data*, D_d: *deciphered data*, A: *access*, M_e: *message*, P_r: *policy*, J: *JSON file*, R_d: *request to data access*, U: *user*, K: *key for deciphering*, A_e: *AES algorithm* $\}$
2: **Set** $P \hookrightarrow J$
3: $R_d \ulcorner J$
4: **if** $R_d \cong A$ into J **then**
5: **Get** D_e from D_b
6: **Use** K and A_e on D_e
7: **return** D_d
8: **else**
9: **return** M_e
10: **end if**

Steps 4-10 show if a user has a right to data, then grant access and get it from the database and decipher it otherwise request is returned an error message as defined in 1:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \tag{1}$$

The encryption step involves the AES algorithm to get ciphertext. The most important part of this algorithm is the key expansion. The encryption consists of multiple rounds depending on the size of the key. And each round has a new key. The routine creates $4x(N_r + 1)$ words, where N_r is the number of rounds. We can use a particular equation to calculate and find keys in each round easily as defined in 2:

$$K[n] : s[i] = k[n-1] : s[i] \oplus k[n] : s[i] \tag{2}$$

where k is the size of the key that consists of 16 bytes and s represent every four bytes of that key.

For s_0 we have to use a particular equation that is different from the above equation as follows: These equation are used to find a key for each round rather than s_0, the key from initial key we have, where $R_{con}[i]$ is the round constant for round i of the key expansion.

$$\begin{aligned} &K[n] = s_0 = k[n-1] : s_0 \\ &\oplus SubByte(k[n-1] : s_3 \\ &\gg 8 \oplus R_{con}[i]) \end{aligned} \tag{3}$$

Here n is a constant which is equal to $0x63$, Arr is initial bytes and x^r helps to find the inverse determined by the following nonlinear equation as in 4:

$$B_{out} = Arr \times x' + n \tag{4}$$

To calculate the number of rounds we use the following mathematical equation as as defined in 5:

$$n_r = \frac{S_k}{S_b} + 6 \tag{5}$$

where S_k is the key size and S_b is the block size.

To implement byte substitution, equation 6 is used:

$$\begin{aligned} &b_i = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \\ &\oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus C_i \end{aligned} \tag{6}$$

where b_i is the ith bit of the given byte and c_i is a bit of the given byte with a specific value.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

Algorithm 3 Intrusion Detection

Input: $\{D_s, L_o\}$ in
Output: $\{M_e, D_s\}$ out

1: **Initialization:** $\{D_s$: *dataset*, D: *data*, D_b: *database*, D_e: *encrypted data*, D_d: *deciphered data*, A: *access*, M_e: *message*, Lo: *data in the log*, U: *user*, P_a: *patterns*, S: *active status*, M_{sd}: *supervised model*, M_{ud}: *unsupervised model*, N_l: *new log* $\}$
2: **Train** M_{sd} and M_{ud} using D_s and L_o
3: **Do**
4: **Apply** M_{sd} on N_l
5: **if** $N_l \cong P_a$ **then**
6: **return** M_e
7: **else**
8: **Apply** M_{ud} on N
9: **end if**
10: **if** $N_l \neq L_o$ **then**
11: **return** M_e
12: **end if**
13: **While** $U \cong S$

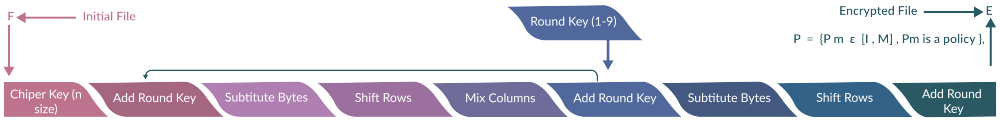


Figure 1. Data encryption with the AES algorithm.

In algorithm 3, step 1 shows the initialization process of used variables. The input and output processes are shown at the beginning of the algorithm. Step 2 shows the process of training our models using a dataset and previous log files.

Steps 3-13 checks the current log of the user against known patterns and previous logs using trained models to check if the behavior of that user is malicious, then return warning message about intrusion to the administrator of the system otherwise continue to monitor while the user is active.

In order to shift the rows, which is the next step in AES algorithm, we have used equation 7:

$$S'_{r,c} = S_{r,(c+shift(r,Nb))modNb}, \quad for\ 0 < r < 4\ and\ 0 \leq c \leq Nb, \tag{7}$$

where the function $shift(r, Nb)$ is used to create the action of moving the byte to lower position and is dependent on the row value itself, s is the byte substitution table, r is row, c is column, Nb is number of bytes.

For mixing the columns, equation 8 is used:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}, \tag{8}$$

which is a fixed polynomial and then used in the following equation for mixing the columns in the matrices, s' is the new byte, $s(x)$ is byte at position x in substitute table and a is byte at position x from initial array table:

Figure 1 shows our data encryption algorithms. As you can observe the initial file will be ciphered using the key. The key will change in each round.

Then the byte substitution is implemented and the rows are shifted as follows:

After this, the column will be mixed and everything starts again. We will use a dataset for intrusion detection. This dataset contains categorical and numerical features with

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

various scales that is why we need to transform it into metric space, i.e. numeric values, and normalize them. Below we demonstrate how we do it.

Each categorical feature expressing n possible categorical values is transformed to a value in Rn using a function f that maps the m-th value of the feature to the m-th component of an n-dimensional vector:

$$f(x_i) = (0, ..., 1, ..., ..., n) \tag{9}$$

where x_i equals to the value of m. As you can see in the formula, the 1 in the braces is at position m. Then we apply on these new features and numeric features scaling function by using their corresponding mean μ and standard deviation σ values:

$$f(x_i) = \frac{x_i - \mu}{\sigma} \tag{10}$$

To create neural network we have used the following formula for one neuron to calculate the hidden layer:

$$h(x) = \sigma\left(w_j + \sum_{i=1}^{n} w_{ij} x_i\right) \tag{11}$$

where σ is a non-linear activation function x_i is the initial i^{th} node and the w is the weight for that particular node. To calculate the output we used the following equation:

$$o = \sum_{j=1}^{n} h_j w_j \tag{12}$$

where h is a hidden layer and w is the weight for that particular hidden layer.

The attributes of each entity define the identity and characteristics of its corresponding entity.

- $A_{req} = \{ReqAttr_i | i \in [1, I]\}$
- $A_{serv} = \{ServAttr_j | j \in [1, j]\}$
- $A_{res} = \{ResAttr_k | k \in [1, K]\}$
- $A_{env} = \{EnvAttr_l | l \in [1, L]\}$, where I,J, K and L represent the maximum number of attributes per entity.

Security policies are defined in the system of the cloud. Each user may have its defined policies. The policy that ABAC supports as a superset of these policies is defined as in 13:

$$P = \{Pm \in [I, M], Pm\ is\ a\ policy\}, \tag{13}$$

Access decision is taken on policy evaluation and the decision is made by the decision function f. P_f is the evaluation function of policy p_n and is defined as follows:

$$Pn_f(A_{req}, A_{serv}, A_{res}, A_{env}) = permit\ or\ deny \tag{14}$$

Polices are evaluated by passing the attributes of the entities to the decision function f, defined as follows.

$$\begin{aligned} &Decision_ABAC = f(Requestor,\ Service \\ &,\ Resource,\ Environment) \\ &P1_f(Requestor)\&P2_f(Service)\& \\ &P3_f(Resource)\&P4_f(Environment), \end{aligned} \tag{15}$$

The encrypted data is sent to the database. When the user requests the data ABAC algorithm will be called. It will use the JSON file, where all the policies and rules are defined.

The attributes of the user, action, and data will be compared. And when the correspondence is found the access will be granted to the user. The AES algorithm again is applied to decipher the encrypted data.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

Figure 2 illustrates data encryption and data access algorithms. The initial key will be transformed to another form using key expansion. This key then will be used to encrypt data using logical XOR.

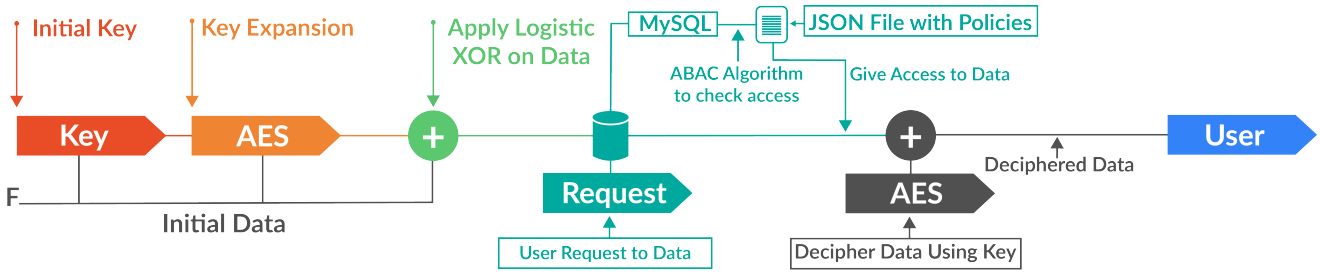


Figure 2. Data encryption and data access algorithms.

5. Implementation and Experimental Results

In this section, the implementation of the mechanism is described. In table 1 shown used materials during the implementation. This mechanism is built on MacOS Mojave operating system using Pycharm IDE in Python 3.4.6 language and IntelliJ IDE in java. Also, JSON file has been used to write policies and rules to implement the ABAC data access algorithm. All of the data and user information are stored in a MySQL database. The existing UNSW-NB15 dataset [49] has been used for IDS implementation. Normal and abnormal network traffic have been generated in a real world test-bed. The abnormal traffic are generated using hacking tools, which consists of nine types of attacks namely: Denial-of-service, Worms, Shellcode, Exploits, Generic, Fuzzers, Backdoors, Reconnaissance, and Analysis. A ground truth table consisting of the nine types of simulated attacks to label the dataset. The dataset input is a set of instances. Each instance contains various data types such as float, integer, binary, and nominal. The instances that belong to one of the nine types of simulated attacks have been labeled as abnormal (i.e., 1) while the normal instances have been labeled as 0. The number of instances (i.e., the rows in 2-dimensional space) in the dataset is 2,540,044. The number of instances that belong to the normal class is 2,218,761 while the number of instances that belong the abnormal class is 321,283. The number of features (i.e., the columns in 2-dimensional space) is 49 representing the network packet fields such as source IP address, destination IP address, source port number, destination port number and so on.

Table 1. Used Materials.

Materials	Description
Platform	Python 3.4.6, Java
Operating system	MacOS Mojave
RAM	8GB
ROM	128GB
Database	MongoDB
Processor	1.8 GHz Intel Core i5
Data set	UNSW-NB15
Policy list	JSON
Environment	Pycharm, IntelliJ IDE

Figure 3 shows the authorized data access case. When the user tries to access data, the algorithm uses user, data, action attributes from the database, as well as the policies from

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

the JSON file to check the privileges and rights. If a user tries to access data for which he has no rights, access will be denied. Also, the IDS system constantly monitors the requests of the user, if it finds the anomaly activity, it will notify the system administrator.

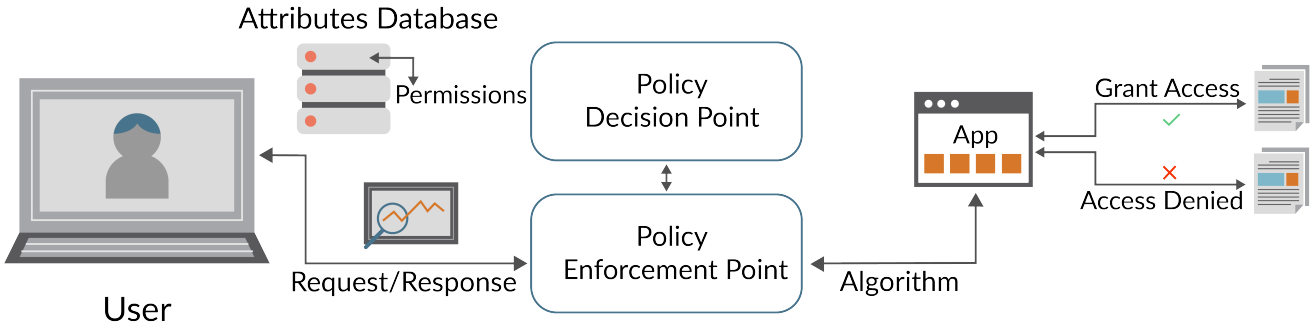


Figure 3. Data access mechanism.

Figure 4 illustrates the case of an attack on the cloud system. The attacker uses the command and control method to get control to cloud providers, who then will produce a huge amount of attacks.

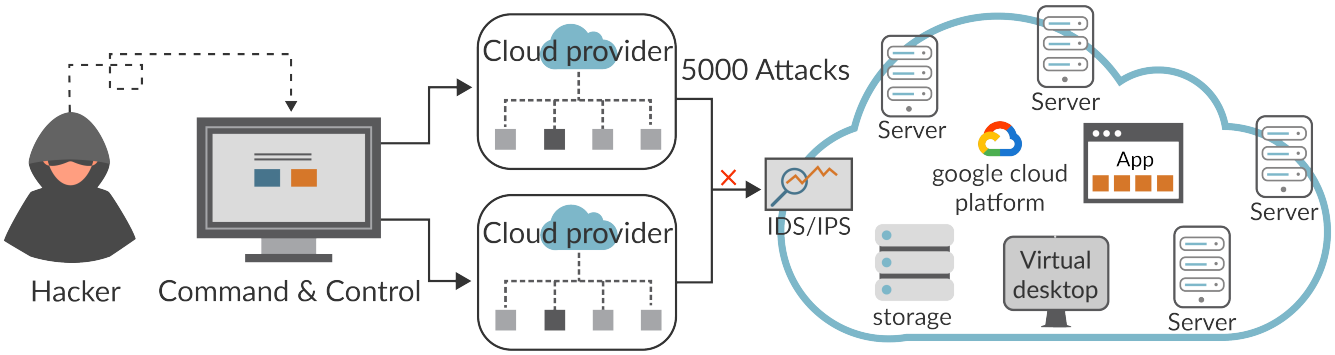


Figure 4. Attacks from outside the cloud.

In this case, 5000 attacks were generated. The cloud computing server itself has IDS and IPS, monitors the network activity. The model which we designed defines the traffic unusual and denies the requests.

IntelliJ IDEA is utilized to implement ABAC algorithm where apache tomcat and maven used to run the project. JSON file for data access. When the user sends a request to the system and it will go to the PermissionEvaluator component, where it will further be delegated to the PolicyEnforcement component, which makes all the decisions based on rules. It will use the PolicyDefinition component to load all the PolicyRules. After loading PolicyEnforcement it will compare the request against the rules, and if the system returns true the access will be granted, otherwise, access is denied.

In Figure 5a we have plotted the graph, which shows the time performance of each request in the administrator account. The project has itself different kinds of requests to the system. They are shown in Table 2. As can be seen the performance for each request quietly different. We have made operations like add, delete, and view, list of users and project tables. Mostly operations for users' tables require more time. The reason for this is the complexity of the table itself, it has many attributes and relationships.

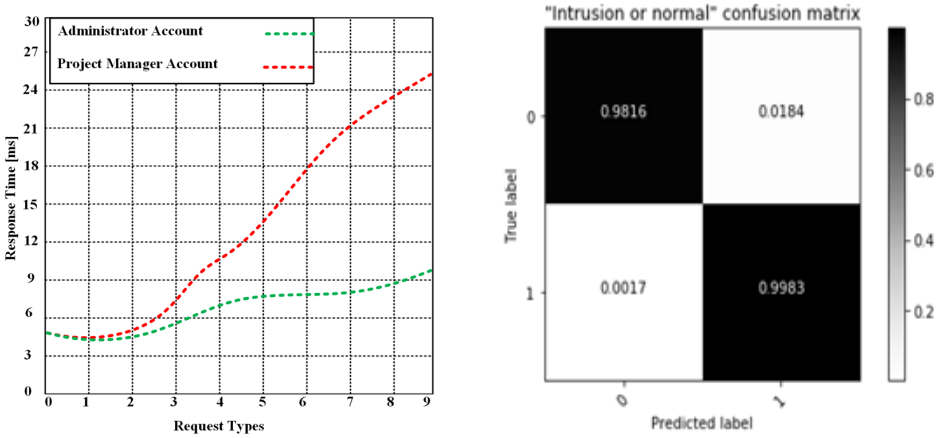
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

Table 2. Tested Requests Types.

1	add project
2	list project
3	view project
4	delete project
5	delete user
6	add user
7	list user

Figure 5a shows different response time. That is because the administrator has more rights to perform some operations. When the administrator makes requests, it involves the database and JSON file, while the project manager just gets a denial and involves only JSON file.

Figure 5b shows the confusion matrix, which shows predicted and real results. As you can see from the below matrix the model predicted quite well, 0.9816 right answers and 0.0017 for normal behavior and 0.9983 right and 0.0184 wrong answers for intrusion behavior.



(**a**) (**b**)

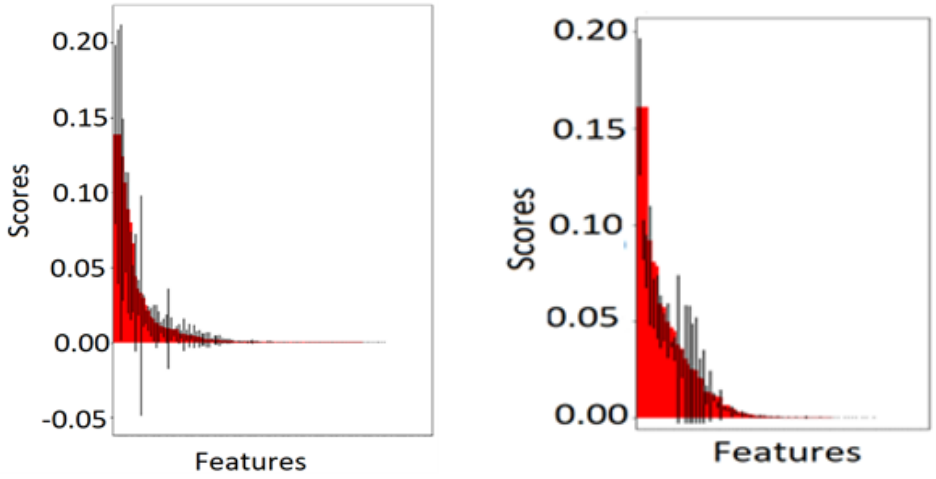
Figure 5. (**a**) Time spent in administrator and project manager accounts. (**b**) Confusion matrix of the classification model.

Figure 6a shows the features of important for Random forest algorithm and Figure 6b demonstrates the features of important for neural network algorithm. These features are importance to choose the only needed attributes to train the model. These chosen features mostly impact to define the intrusion to the system.

By doing so, we have increased the accuracy of the model. By testing this algorithm we confirm that the algorithm works correctly. All the access decisions are based on the rules defined within the default-policy (.json file). All users are defined in the Memory_User_Details_Service file. HIDS was developed using the random forest and feed-forward neural network. The system uses the random forest that classifies data to normal or malicious data. The resulted information then further used to train a neural network to classify the attack data based on the different attack categories.

Figure 7a displays the Receiver operating characteristic (ROC) curve of the classification model. It shows the ability of the classification model to diagnose the given request. It is based on confusion matrix results. In summary, it shows the accuracy of the trained model. The results show that ROC is equal to 0.9962. This is a good result, that can

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1

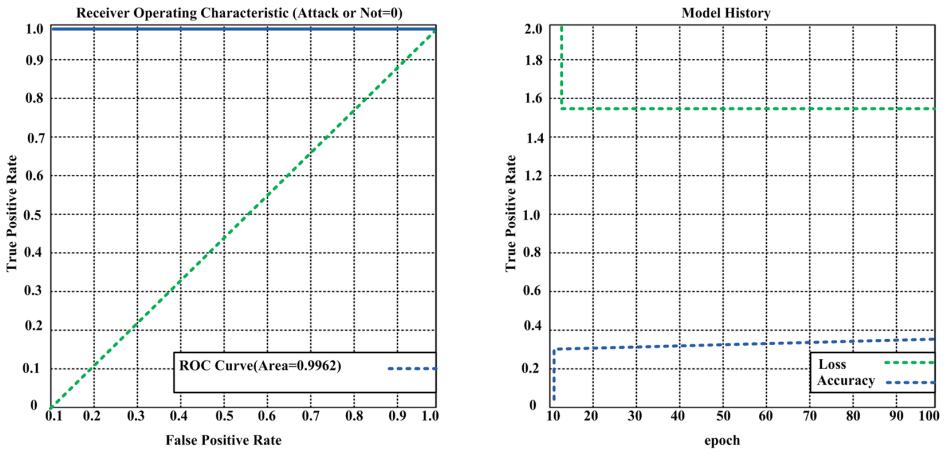


(**a**) (**b**)

Figure 6. (**a**) Feature importance for Random Forest. (**b**) Feature importance for Neural Network.

be used to predict, whether the user is authorized or not by integrating it into different environments.

Figure 7b demonstrates the history of our neural network model. As it has been observed that the loss is reduced, while the accuracy of the model was increased as the number of epochs was increased. The accuracy increased 0.36% and loss reduced approximately 0.47%. Based on the result, it can be claimed that proposed model produced higher performance.



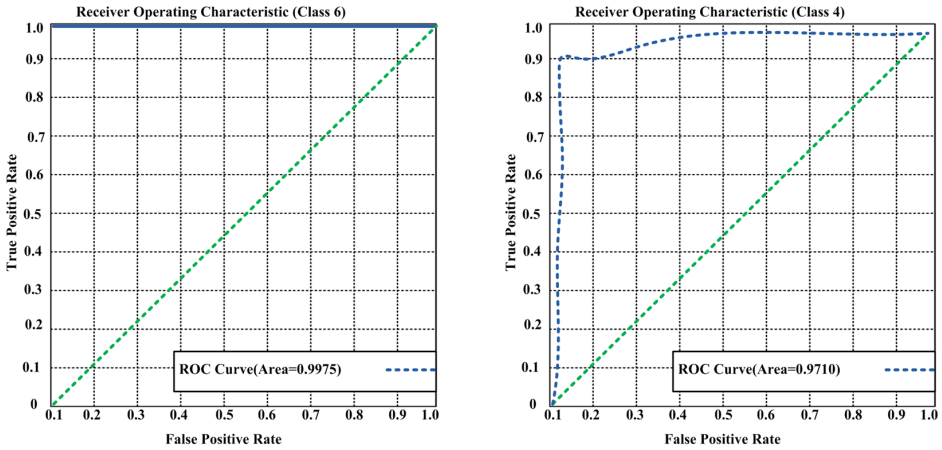
(**a**) (**b**)

Figure 7. (**a**) ROC curve of the classification model. (**b**) Neural network Model history.

Figure 8a is illustrates the ROC curve of our neural network model when the class is equal to 6. The Receiver operating characteristic curve shows 0.9975 of accuracy. There also classes 4, 2, and 0 which is considered to be important in the feature importance step. We have shown only classes 6 and 4 for comparison only.

Figure 8b illustrates the ROC curve of our neural network model when the class is equal to 4. The Receiver operating characteristic curve shows 0.9710 of accuracy for this class a high importance value in the previous step. Based on the results, we observe that Figures 8a and8b use a different class. Thus, they get different accuracy ratios. If a class is higher, then higher accuracy can be obtained.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 December 2021 doi:10.20944/preprints202112.0068.v1



(**a**) (**b**)

Figure 8. (**a**) ROC curve for class 6. (**b**) ROC curve for class 4.

6. Discussion of Result

Different from the existing methods in the literature, our method applies access control to reduce the possibility of unauthorized data access, enhances the accuracy of intrusion detection system, and provide more protection by encrypting the data in the cloud. Table 3 lists the security capabilities of our method when compared to others. Compared to the other methods that try to solve a part of the solution, our method is a complete solution that prevents unauthorized access of cloud computing.

Table 3. Comparisons with extsting methods.

Method	IDS	Encryption	Access control
[43,44]	✓	-	-
[17,32,40]	-	-	✓
[39]	-	✓	✓
Ours	✓	✓	✓

We also chose to adopt a light symmetric encryption algorithm (i.e., AES) to avoid the expensive computations that other public cryptography based related works have. Public key encryption are slower than private key encryption and some solutions suggested to use cloud servers for decryption which is impractical in some situations such IoT environments.

Moreover, unlike some related works that use rule-based approaches (i.e., requires an expert who is responsible of designing the handy-crafted IDS rules) The proposed IDS is automated (i.e., based on machine learning) that achieved decent results. It has achieved high accuracy as indicated in the experimental results section.

7. Conclusion

The HUDH scheme is introduced that combines three state-of-the-art algorithms (AES, ABAC and HIDS) for improving data access control in cloud computing. ABAC and AES algorithms are implemented using JSON file. The proposed HUDH has significantly improved the data security and user authentication. The intrusion detection accuracy is improved using the features of the neural network algorithm. One of the advantages of using the proposed algorithm is to reduce the possibility of unauthorized data access. We implemented ABAC and AES algorithms in Java and HIDS on python. HIDS algorithm gets features from two algorithms: First, the Random Forest algorithm is used for the selection of important features. Second, the neural network model is used to train the data.

The HIDS further detects intrusion and the accuracy of class-4 and class-6 has been determined 0.9710 and 0.9975 respectively. Moreover, the proposed HUDH scheme can enable the data owner to delegate most of the computation overhead to powerful cloud servers. Confidentiality of user access privilege and user secret key accountability are achieved. Formal security proofs show that the proposed scheme is secure under standard cryptographic models. In the future, we will compare the HUDH scheme with similar types of state-of-the-art mechanisms using different Quality of service parameters.

Author Contributions: A.R. and S.N., conceptualization, writing, idea proposal, methodology and results; B.A. and M.A., conceptualization, draft preparation, editing and visualization; A.M., writing and reviewing; A.A. draft preparation, editing and reviewing. All authors have read and agreed to this version of the manuscript.

Funding: This work was partially supported by the Sensor Networks and Cellular System (SNCS) Research Center under Grant 1442-002

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Taif University Researchers Supporting Project number (TURSP-2020/302), Taif University, Taif, Saudi Arabia. The authors gratefully acknowledge the support of SNCS Research Center at the University of Tabuk, Saudi Arabia. In addition, the authors would like to thank the deanship of scientific research at Shaqra University for supporting this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Borylo, P., Tornatore, M., Jaglarz, P., Shahriar, N., Chołda, P., Boutaba, R. Latency and energy-aware provisioning of network slices in cloud networks. *Computer Communications* **2020**, *157*, 1-19.
2. Razaque, A., Frej, M. B. H., Alotaibi, B., Alotaibi, M. Privacy Preservation Models for Third-Party Auditor over Cloud Computing: A Survey. *Electronics* **2021**, *10*, (21), 2721.
3. Kassab, W. A., Darabkh, K. A. A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations. *Journal of Network and Computer Applications* **2020**, *163*, 102663.
4. Sun, P. Security and privacy protection in cloud computing: Discussions and challenges. *Journal of Network and Computer Applications* **2020**, *160*, 102642.
5. Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., Inácio, P. R. (2014). Security issues in cloud environments: a survey. *International Journal of Information Security*, **2014** *13* (2), 113-170.
6. Guan, S., Niu, S. (2021). Stability-Based Controller Design of Cloud Control System With Uncertainties. *IEEE Access* **2021**, *9*, 29056-29070.
7. Namasudra, S. Cloud computing: A new era. *Journal of Fundamental and Applied Sciences* **2018**, *10*(2).
8. Amani, M., Ghorbanian, A., Ahmadi, S. A., Kakooei, M., Moghimi, A., Mirmazloumi, S. M., ... Brisco, B. Google earth engine cloud computing platform for remote sensing big data applications: A comprehensive review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2020**, *13*, 5326-5350.
9. Amazon, E. C. (2015). Amazon web services. Available online: http://aws. amazon. com/es/ec2/(November 2012), 39.
10. Cloud, A. E. C. (2011). Amazon web services. Retrieved November, 9(2011), 2011.
11. Martin, R. (2008). IBM brings cloud computing to earth with massive new data centers. InformationWeek.
12. I. Google, "Google app engine," Available online: https://searchaws.techtarget.com/definition/Google-App- Engine(2014), 2014.
13. Kulkarni, G. (2012). Cloud computing-software as service. *International Journal of Cloud Computing and Services Science* **2012**, *1* (1), 11.
14. Rai, R., Sahoo, G., Mehfuz, S. (2013). Securing software as a service model of cloud computing: Issues and solutions. *arXiv preprint* **2013** arXiv:1309.2426.
15. Neubert, B. C. M. (2018). Valuation of a Saas Company: A Case Study of Salesforce. Com. INNOVATION MANAGEMENT, *ENTREPRENEURSHIP AND SUSTAINABILITY* **2018**, *166*.
16. E. Azure, "Azure web services," Available online: https://azure.microsoft.com/en-us/(December, 2021), 2021.
17. Yu, S., Wang, C., Ren, K., Lou, W.. Achieving secure, scalable, and fine-grained data access control in cloud computing. In 2010 Proceedings IEEE INFOCOM (2010, March), (pp. 1-9). Ieee.
18. Wan, Z., Deng, R. H. HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE transactions on information forensics and security* **2011**, *7*(2), 743-754.

19. Wang, G., Liu, Q., Wu, J. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In Proceedings of the 17th ACM conference on Computer and communications security (2010, October) (pp. 735-737).
20. Razaque, A., Jararweh, Y., Alotaibi, B., Alotaibi, M., Hariri, S., Almiani, M. (2022). Energy-efficient and secure mobile fog-based cloud for the Internet of Things. *Future Generation Computer Systems* **2022**, *127*, 1-13.
21. Singh, J., Dhiman, G. A survey on cloud computing approaches. *Materials Today: Proceedings.* 2021.
22. Almusaylim, Z. A., Jhanjhi, N. Z. (2020). Comprehensive review: Privacy protection of user in location-aware services of mobile cloud computing. *Wireless Personal Communications* **2020**, *111* (1), 541-564.
23. Masud, M., Gaba, G. S., Choudhary, K., Alroobaea, R., Hossain, M. S. (2021). A robust and lightweight secure access scheme for cloud based E-healthcare services. *Peer-to-peer Networking and Applications* **2021** 1-15.
24. Razaque, A., Amsaad, F., Hariri, S., Almasri, M., Rizvi, S. S., Frej, M. B. H. Enhanced grey risk assessment model for support of cloud service provider. *IEEE Access* **2020**, *8*, 80812-80826.
25. Razaque, A., Almiani, M., Khan, M. J., Magableh, B., Al-Dmour, A., Al-Rahayfeh, A. Fuzzy-gra trust model for cloud risk management. In 2019 Sixth International Conference on Software Defined Systems (SDS), (2019, June)., (pp. 179-185). IEEE.
26. Li, M., Yu, S., Ren, K., Lou, W. . Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In International conference on security and privacy in communication systems (2010, September), (pp. 89-106). Springer, Berlin, Heidelberg.
27. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D. The eucalyptus open-source cloud-computing system. In 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (2009, May), (pp. 124-131). IEEE.
28. Khan, A. R. Access control in cloud computing environment. *ARPN Journal of Engineering and Applied Sciences* **2012** *7*(5), 613-615.
29. Zissis, D., Lekkas, D. Addressing cloud computing security issues. *Future Generation computer systems* **2012**, *28*(3), 583-592.
30. Hota, C., Sanka, S., Rajarajan, M., Nair, S. K. Capability-based cryptographic data access control in cloud computing. *International Journal of Advanced Networking and Applications*, **2011**, *3*(3), 1152-1161.
31. Namasudra, S.Taxonomy of DNA-based security models. *In Advances of DNA computing in cryptography* **2018** (pp. 37-52). Chapman and Hall/CRC.
32. Chinnasamy, P., Deepalakshmi, P. HCAC-EHR: hybrid cryptographic access control for secure EHR retrieval in healthcare cloud. *Journal of Ambient Intelligence and Humanized Computing* **2021**, 1-19.
33. Kumar, K. S., Nair, S. A. H., Roy, D. G., Rajalingam, B., Kumar, R. S. (2021). Security and privacy-aware Artificial Intrusion Detection System using Federated Machine Learning. *Computers & Electrical Engineering* **2021**, *96*, 107440.
34. Han, S., Han, K., Zhang, S. A data sharing protocol to minimize security and privacy risks of cloud storage in big data era. *IEEE Access* **2019**, *7*, 60290-60298.
35. Razaque, A., Rizvi, S. S. Privacy preserving model: a new scheme for auditing cloud stakeholders. *Journal of Cloud Computing* **2017**, 6(1), 1-17.
36. Yang, K., Jia, X. ABAC: Attribute-based access control. *In Security for cloud storage systems* **2014**, (pp. 39-58). Springer, New York, NY.
37. Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., Razaque, A. Deep recurrent neural network for IoT intrusion detection system.*Simulation Modelling Practice and Theory* **2020**, *101*, 102031.
38. Zhang, Z., Zeng, P., Pan, B., Choo, K. K. R. Large-universe attribute-based encryption with public traceability for cloud storage. *IEEE Internet of Things Journal* **2020**, *7*(10), 10314-10323.
39. Pourvahab, M., Ekbatanifard, G. Digital forensics architecture for evidence collection and provenance preservation in IaaS cloud environment using SDN and blockchain technology. *IEEE Access* **2019**, *7*, 153349-153364.
40. Riad, K., Hamza, R., Yan, H. Sensitive and energetic IoT access control for managing cloud electronic health records. *IEEE Access* **2019**, *7*, 86384-86393.
41. Hahn, C., Kim, J., Kwon, H., Hur, J.. Efficient iot management with resilience to unauthorized access to cloud storage. *IEEE Transactions on Cloud Computing* **2020**.
42. Alkadi, O., Moustafa, N., Turnbull, B., Choo, K. K. R. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. *IEEE Internet of Things Journal* **2020** *8*(12), 9463-9472.
43. Kimmel, J. C., Mcdole, A. D., Abdelsalam, M., Gupta, M., Sandhu, R.. Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure. *IEEE Access* **2020**, *9*, 68066-68080.
44. Abdelsalam, M., Krishnan, R., Huang, Y., Sandhu, R. . Malware detection in cloud infrastructures using convolutional neural networks. In 2018 IEEE 11th International conference on cloud computing (CLOUD) (2018, July), (pp. 162-169). IEEE.
45. Al Makdi, K., Sheldon, F. T., Hussein, A. A. Trusted Security Model for IDS Using Deep Learning. In 2020 3rd International Conference on Signal Processing and Information Security (ICSPIS), (2020, November), (pp. 1-4). IEEE.
46. Namasudra, S. An improved attribute-based encryption technique towards the data security in cloud computing. *Concurrency and Computation: Practice and Experience*, **2019** *31* (3), e4364.
47. Namasudra, S., Roy, P. PpBAC: popularity based access control model for cloud computing. *Journal of Organizational and End User Computing (JOEUC)* **2018**, *30* (4), 14-31.

48. Namasudra, S., Roy, P., Balusamy, B., Vijayakumar, P. Data accessing based on the popularity value for cloud computing. In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) (2017, March), (pp. 1-6). IEEE.
49. Moustafa, N., Slay, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (2015, November), (pp. 1-6). IEEE.