Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.7 n.29, p. 36-46. May/Jun, 2021

DOI: https://doi.org/10.5935/jetia.v7i29.750

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

A SYMBOLIC ATTRIBUTE-BASED ACCESS CONTROL MODEL FOR DATA SECURITY IN THE CLOUD

Iyabo Felicia Oyeyinka*[1], Sunday Idowu[2] and Afolashade Kuyoro[3]

[1] Centre for Information Technology Management, Yaba College of Technology, Yaba, Lagos State, Nigeria.
[2, 3] Computer Science Department, School of Computing and Engineering Sciences, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

[1] http://orcid.org/0000-0001-5826-0575, [2] http://orcid.org/0000-0002-0013-0265, [3] http://orcid.org/0000-0001-7235-7744

E-mail: *oyeyinka0467@pg.babcock.edu.ng *(kemi.oyeyinka@yabatech.edu.ng), idowus@babcock.edu.ng, kuyoros@babcock.edu.ng

ARTICLE INFO

Article History
Received: May 05th, 2021
Accepted: June 21th, 2021
Published: June 30th, 2021

Keywords:
Access control,
Authentication,
Cloud Computing,
Data Security,
Encryption.

ABSTRACT

There have been several attempts made in literature to develop access control techniques to stem data security problems. Many of these techniques had been found to have one deficiency or other. Hence, this study developed a Symbolic Attribute-Based Access Control (SABAC) system for data security in the cloud service environment. SABAC system was implemented by developing Hash-tag Symbol Authentication (HSA) algorithm using the Message Digest-5 encryption. SABAC utilizes a 3-Tier continuous authentication method by combining the use of username and password, HSA code, and real-time image monitoring and verification. HSA code is generated by combining 5-tuple user attributes and the string generated from the user's image using Obfuscation Technique. The concatenated string is converted to hexadecimal which serves as input to MD5 to produces a unique HSA code. SABAC was evaluated using three major security metrics of confidentiality, integrity, and avail-ability. The result of security metrics tests showed a confidence level of 99.993%, integrity threshold of 99.998%, and availability throughput of 150 users/second. This implies that SABAC is highly efficient for cloud data security. It shows that hackers would find it impossible to match any fake identity with valid HSA in the database. The study concluded that SABAC could be used for access control in a cloud environment for assuring data security. It was recommended that the SABAC system should be adopted by Cloud Solution Providers and Security Specialists.

I. INTRODUCTION

In the conventional form of computing, the cost of setting up computer services is very high, including infrastructure maintenance, technical training, software licensing, and upgrading costs. The way information communication technology systems are managed has changed with improved Technology. This has resulted in a dramatic reduction in the cost of computing services and operation. Furthermore, while the cost of running IT services and many other related IT operational activities are reduced, the fear of data theft, impersonation, and eavesdropping, as well as other forms of cybercrime, remains a common drawback [1]. Security of data and resource has continuously been a major challenge of computing services. Cybercrime is a serious threat to computing technology, people, organizations, and nations, according to a survey published by [2], and the situation is compounded by data loss. According to [3] in the Verizon Business Data Bridge Investigation report, cyber vulnerability and attacks accounted for 43% of breaches on the web in 2020. It was shown in the report that out of these breaches, hacking accounted for 45%, social attacks 16% and human error caused 22% of breaches. This is more than double the proportion in 2019.

Therefore, a successful solution to the problems of computing vulnerability needs to keep data out of the reach of unauthorized access. One way of doing this is by monitoring people's rights by using a secure method of access control through a reliable system of authentication. In access control, it is a requirement to check the users' rights before granting permission. At any given time, access is limited or granted to the approved user and the implementation of access management techniques must be

capable of detecting any form of impersonation or infringement. Many access control techniques has been proposed, these include: Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-Based Access Control (RBAC), Relationship-Based Access Control (ReBAC), Attribute-Based Access Control (ABAC) etc. ABAC is effective and efficient for big companies or big data [4]. All the existing Access Control techniques, out of which RBAC and ABAC are the most popular, have shortcomings fitting into the data security problems in the cloud service environment. RBAC has the problem of role explosion which happens when an organization's roles outnumber its actual users, and role changes may cause the authorization associated with each role to be modified or removed. There is a limitation to the volume of data this tool can handle hence it can only be used for a private cloud. ABAC is policy-based but neutral in that it can be used in many access control strategies to communicate various types of policies and the decision-making capability is very dynamic, which is a very significant benefit. ABAC uses the attributes of the user and resource for decision making. However, user attributes are also at risk of being compromised by intruders due to the multiple rules involved in permitting the user. It is not specifically defined what attribute should be, the effect of altering attribute is not known. Besides, the policy's implementation of ABAC is complex and difficult due to the fact that it is rule-based. It requires a lot of policies for each of the constraints to be implemented, such as object, subject and other underlined entities. In addition, there was no method to classify and monitor users' attributes as they travel through the system. ABAC leveraged on authorization more than authentication [5] hence we proposed a new system that is authentication-based.

In this paper a Symbolic Attribute-Based Access Control (SABAC) model is presented. It aims at protecting users' attributes so that their data is not exposed to intruders. To accomplish this, a symbolic code is generated by concatenating the image of the user with other attributes to produce a unique hashtag code for each users. In SABAC technology, the picture is a very important attribute that has been captured and encrypted to compute the hashtag using MD-5. The model is novel because the encrypted image is converted to code that hides the user's identity from an unauthorized person.

The remaining of this paper is organized as follows: Section 2 describes the background of the studies and related works, section 3 presents the design of SABAC and section 4 concludes the paper.

II. BACKGROUND AND RELATED WORKS

II.1 BACKGROUND

The overview of the previous works on access control indicated that ABAC is reliable and more efficient than others in literature but its shortcoming makes it not perfect for use in the cloud. So far, any appropriate formal specification for ABAC and its operational functionality has not been recorded. ABAC is characterized by NIST as an access management system where approvals are granted or refused on the basis of attributes assigned to the subjects, objects, environmental, and a collection of policy specifying the attributes [6]. The ABAC model was strongly acknowledged to be in development process and still have varying viewpoints on it [7].

ABAC is made up of Constraint, Permission, Object, Subjects and Users. Each of these has an attribute attached; Object Attribute (OA), Subject Attribute (SU), User Attribute (UA) [8]. The terms attributes are applied differently by organizations depending on what the system is meant to do. In some cases the ABAC model consists of entities called subject; that can be referred to as device, program, mechanism, user etc. Likewise, the object can be server, network, archive, and so on. Whereas, the context attributes are the environmental factors and constraint are the policies. The entities depicted as user will generally represent a person, a system representation of user assigned a selected session or time-frame could be also be a subject. An object could be a resource to be accessed by a user. The context may be when the access is made, location, the device actions, etc. Precise user characteristics include names, date of birth, rental day, address, telephone number, occupational title, identification number, etc. The addresses of the Internet Protocol, Global Positioning System are the values associated with location. The user needs to be assigned attributes to identify permission to access the services.

ABAC permits a lot of flexibility and individualization in access control management by combining the user attribute to data for decision making on the resources. This quality is more reliable to exploit in cloud service environment than using users' role or hierarchy as in RBAC. ABAC, provides a lot of management for the protection of data than the RBAC. It's built towards handling large information like what it's obtainable in the cloud computing [9]. The ABAC model works through a combination of authentication and access control authorization, which seems too cumbersome and poorly fixed. Authorization governs the approval of users to access resources dynamically with a number of rules guiding authorization. User will solely be permissible to access objects that its attributes are confirmed to align with the configured policies. This should be verified to meet the requirements of authorization after approval. Additionally, constraint specification is a sensitive aspect of achieving viable and robust access policy in a company that uses ABAC and it causes implementation difficulty as a result of several attributes to be specified for the entities concerned. That suggests there are several policies to be specified in ABAC unlike in RBAC that make use of the singular role of the user in the organization.

Any security model's behaviour dictates its simplicity. [10] claimed that ABAC's policy requirements are too many and that implementations is also become tasking. Also, ABAC is difficult to evaluate because of different characteristics and policies. If a system is not easy enough to be used by anybody without strict guidelines, it may be labelled as weak or not successful. ABAC focuses mainly on what the user is doing with the resource, which is authorization and mellows down on authentication since the system is more on policy [11].

II.2 RELATED WORKS

In their article Cross Bread Role-based Access Control for Extended Security at Azure in Cloud Computing. The proposed Advanced RBAC Architecture is a form of ontological definition which keeps a records of backup data. The log and user limit per position is sent to the server on the cloud. There is no backup in the previous framework that can lead to data security risks in the event of data failure. These new features have the advantage of improving cloud storage protection and avoiding data loss. In terms of the number of roles per user, the architecture addressed in this work has a constraint [12].

In "A New Semantic Role-Based Access Control Model for Cloud Computing", stated that the present role-based access control is user-centric, if indeed the user has no permission, the demanded access function would be denied. Meanwhile the semantic model implemented functions and responsibilities of the request of the user. The model contains the broker, the user, and

the knowledge-based layers respectively. The user layer does the job of dispatching the request to the broker layer. It is necessary to note that user's requests cannot be rejected under this system, rather they should be re-organized to find suitable roles and positions that suit the user's purpose. This will be forwarded to the authorization agent, who will submit the assignments of the responsibilities and functions to the knowledge-based layer. The database consist of SPARQL, which is responsible for running a query for the user's discovered roles and functions once it is applicable to the user's permission request. This layer also consists of a graph of ontology that maps the permissions, functions, and features. The graph showed the direct relationship at the request level between roles and functions as opposed to the relationship at the permission level between roles and functions. Although this model is very flexible and versatile for cloud services, but searching for a matching position and feature that suits the request of the user may be time-consuming [13].

Study on ABAC and the implementation of IaaS models was carried out by [5]. This was based on the current model and the shortcomings of the role explosion in RBAC. The inconveniences of RBAC necessitated the development of the ABAC model. It is made up three phases; harmonization of the DAC, MAC, and RBAC features for ABAC, Second, in ABAC, a new model was developed to manage the dynamic characteristics of RBAC and its extension. Finally, to handle user attributes, an administrative model was developed. The generalized user-role assignment model is called the administrative model (GURA). This model was poor because if membership laws for permission are not met, an automated modification for an automated features selection should be provided, but such a requirement was not obtainable in the current ABAC.

The [14] invented and developed Cipher-text Policy-Based Encryption (CP-ABE) to manage secure access in his work, Toward Successful Access Control using Attributes and Pseudo Roles. Two-step access controls were developed and validated through Bilayer Access Control (BLAC) approach. In BLAC, level one confirms if user has made requests to access BLAC pseudo-roles. The level two tested the rules for further access restrictions within the related BLAC policies. BLAC, therefore, makes good use of attributes while retaining the benefits of RBAC. One problem noted in this study is that the user's right to privacy can be exposed because several characteristics are used in the process of requesting for access. Also, due to the repetitive nature of the policies, user's privacy can be exposed to attack. In BLAC, the attributes of the required resources and the requester are forwarded to the access Decision agent on the remote server. BLAC model was still prone to attack even with the policies been stored remotely.

The [15], concentrating on infrastructure-as-a-service, coordinated in the 4th yearly report for state of cloud adoption 2015. The survey asked 930 IT specialists about the use of cloud computing and associated systems. Participants were drawn from project manager to managers and experts who have represented organizations across various sectors. The report revealed the need for wider knowledge and proper education on cloud technology. And consumers should be more enlightened about the fear of losing their data on account of company folding-up to the cloud owners.

In his Ph.D. thesis, Cryptographic Enforcement of Attribute-Based Authentication, [16]. There was an enumeration of the distinction between the various ABA systems. The investigator also tracked the numerous authentication methods in literature that were listed as traceable and untraceable ABA. These elements represent new fields of information security for researchers.

In the scheme of [17] four stages of its authentication procedures used stationary probability. These are: setup, registration phase, phase of authentication, and phase of changing passwords. Because of the normal username and password used, this work has restrictions that could be prone to modification by an unauthorized user.

In "The Design of Hybrid Cloud Migration Techniques", reviewed some recent benefit of cloud computing. Organizations have been motivated by enhancement and cost effectiveness to migrate their data, software, and other material into the cloud. The issues concerning the rate at which cloud was adopted was discussed. An effective way for information to be transferred to the cloud were suggested. Improved protection and access management methods, ease of use of migration software, as well as improved monitoring tools were proposed as areas of further study [18].

In their paper, "Authentication Techniques in Cloud Computing: A Review", [19] presented the different techniques of authentication. It says that if user data is not secure, the cloud has lost its importance. The protection of data in the cloud is very significant. Username and password was among the different techniques listed but turned out to be the least effective of the techniques discussed and is expected to become increasingly irrelevant since it cannot cope with current security challenges on the cloud.

The [20], proposed the Advanced Encryption Standard (AES) Data Encryption versus Decryption Algorithm. AES is among the best methods used for encrypting and decrypting data in the field of cryptography. He said there was no documentation as to where this algorithm could be breached by hackers. It operate in the following major three sizes such as 128, 192, and 256 bits. The block consist of 128-bit cipher size block each.

This research, "A User-Centric Access Management System for Cloud Computing" by [21] showed the concerns users have on the protection of their migrated data and the thought of losing it. It introduced an access management strategy that invoke PKI for user to secure their products such that the security challenges can be addressed. The PKI techniques work for small data, if you use it with big data, it can make your computation problems worse. Secondly, the certificate authority that converts it into a digital certificate by digitally signing the public key will issue a false certificate by fooling users into submitting their data to incorrect certificate.

Between 1989 and 1992, Ron Rivest modeled many types of MD. MD2, MD4, and MD5 are the families of message digests. In 1990, the MD4 algorithm was developed for its computational velocity in software processing. The message is divided into 512-bit consist blocks and produces a digest length of 128 bits. It is also vulnerable to collision attacks, but it generates more resistance than MD2 [22]. Rivest created an updated version called Message Digest 5 (MD5) in 1992 to replace the earlier MD4 because of security problems [23]. It divides the message into 512-bit blocks and, like MD4, generates a 128-bit length digest. If there is an unintended corruption, it is mainly used to verify data integrity. Collision attacks and pre-image attacks are also vulnerable [24].

In ABAC, the following issues were identified which necessitated the design of SABAC;

- Attributes are easy to impersonate,
- Attributes lack standard,
- ABAC is policy-based and it is complex to implement
- and lack of monitoring mechanism for attributes' identification.

Therefore, there is need to build an access control model that is more stable and user-friendly.

III. DESIGN OF SABAC

There are two main aspects of access control: authentication and authorization. This study concentrated on authentication. An authentication tool named Symbolic Attribute-Based Access Control (SABAC) was developed using The Message Digest-5 (MD5) encryption method to produce Hashtag Symbolic Authentication (HSA); a multi-factor, image-based access control mechanism to enhance the ABAC model. Confidentiality, integrity, and availability are the security metrics used for assessing SABAC performance. SABAC is a tripartite model that incorporates the following principles;

(i) Integrity: the approved users are granted access by combining user attributes and the HSA feature.

(ii) Confidentiality: user data was encrypted and decrypted to ensure confidentiality

(iii) Availability: This was accomplished by maintaining a time the data need achieved.

The MD5 encryption algorithm was used to build the HSA framework. This produces a code that is unique for every user. Each facial image used in the authentication process has the unique code attached to it. The SABAC model was developed and improved with the incorporation of Message Digest 5 (MD5) encryption algorithm to produce HSA key. The following principles was used to generate the key; what you have, what you know, and the extension of what you have. The attributes of the users; last name, first name, and picture are what you know, while what you have is the password. These entities have been joined together using the MD5 encryption to produce the HSA key to make an extension of what you have.

Several stages of adding and padding bits substantially made up MD5 algorithm. Depending on the message length, it takes many rounds of iteration to totally transmit message. The longer the message is, the more rounds for each block of message to be processed. In certain other cases, the bit length may be shorter than the expected norm to calculate the MD5, padding may be required for such bit length. To build the standard bit, a 64-bit representation must be inserted. The MD5 storage and process is stored in a buffer [25]. The steps for carrying out the MD5 algorithm are the following: Append padding bit, append representation of 64 bits, Initialize the MD buffer, process every block, and output. Due to the use of three-way authentication, with the symbolic strategy, an attacker cannot readily break into the cloud resources. The HSA code is far too lengthy to assume, and even if the names were guessed, the image feature could not be fabricated. None of the existing systems in literature used an image to produce a key for authentication systems as a major requirement. SABAC is a multi-faceted enhanced technology that incorporates the uniqueness of the picture of the user to ensure that the confidentiality, integrity, and availability of the cloud service are maintained during data access.

The advent of cloud computing has led to several security issues and a great deal of research in the field of cloud security is needed. In the cloud computing service environment, the SABAC model is a solution to these cloud security problems and can prevent many cyber-crime attacks and other security challenges. With SABAC, threats such as eavesdropping, denial of service attack, insider attack, outsider attack, passwords forgery, impersonation etc. are prevented. Refer to Figure 1 for the flow diagram of the SABAC conceptual model. SABAC's conceptual framework is a process that continues without a break. HSA is an important component for tracking authorized users in all phases of SABAC authentication system. HSA acts as a booster which needs to show often while the user access resources. This indicates that the user is authenticated. The SABAC model was developed for cloud systems and is deployed using Infrastructure as a Service (IaaS).

Please note that in the SABAC authentication process, the user image is converted to create the HSA key. A series of iteration processes are repeated when the HSA key is created. Each time a user attempts to log into the system, the symbolic key is compared to the content of the hash table for that user. Figure1 describes the five phases of SABAC.

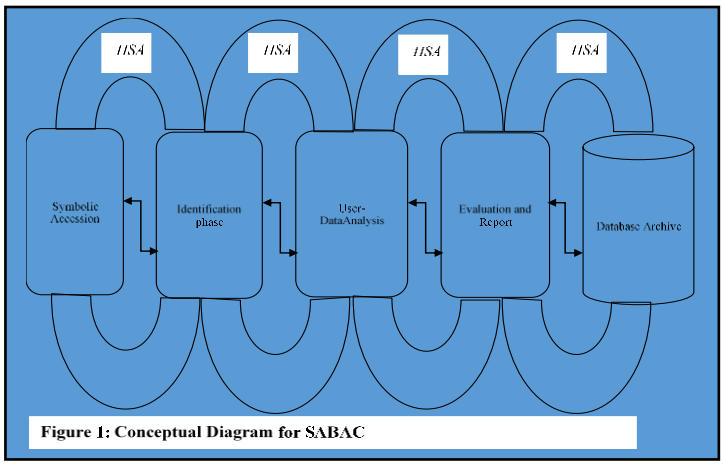

Figure 1: Conceptual Model of SABAC.
Source: Authors, (2020).

The phases used in this model are the symbolic accession process, the identification phase, the analysis of the user's data phase, the report evaluation phase, and the data archive phase. These five stages have an HSA that applies to it, and the transition to the fifth stage includes archiving the users' information.

The symbolic accession process, which involves user registration and identity capture, is the first phase. It is presumed that while some users are new, some may have registered with the owner of the resource. Two processes are involved in the Symbolic Accession phase; new user registration and signing in of existing users. In new user registration, the device prompts the user to supply his/her last name, first name, username and password. To generate the HSA key. The data entered by the user are acknowledged and saved in the repository. To complete the registration process, for new users, the system undertakes the following steps:

i. The user is instructed to take a photograph. The device's camera is turned on, and a picture of the user is taken.

ii. The module generates an image-id for the snapped image. The HSA code is generated as follow:
   - The name of the user such as first-name, other-names, and last-name are saved as 64-bits string.
   - The user's image is translated to string of 64-bit.
   - Every attribute that have been collected are concatenated and converted to hexadecimal values.
   - To produce a hash-tag key an MD5 encryption is applied.

iii. The hash-tag key genereated is store in database for reference and comparison next time the user intends to use the system.

The second process is the confirmation of the existing user profile. The Login procedure for registered individual to get entrance to the system includes the following:

i. The system prompts user to enter username and password.
ii. User's picture is automatically snapped by the system camera.
iii. The data stored in the archive is retrieved and compared with the new ones; the latest picture must be identical to what is in the archive.
iv. A new image-id is created, and the process is compared to the current image-id; the image's similarity and confidence threshold are calculated and saved in the database. If all of the information is identical to that in the database, access is granted; otherwise, access is denied.

Identification phase is the second phase. During this phase, the system serves as a trusted authority. The identification process identifies all the entities that produced the symbolic elements. The trusted authority will do a thorough check of the attribute combined to formed the HSA, and then connect them to the HSA key. The Symbolic portion is transformed for visual confirmation and readability. Since the HSA key is linked to the user image stored in the database, adding it to this stage improves the user's traceability. This distinguishes the work because the symbolic accession process ensures that the user attributes are captured correctly and encrypted. The identification phase ensures access is given to the authorized user by ensuring simple verification process. For example, a session can shut down unexpectedly, thereby allowing an attacker to login through the last login session that was not properly turned off. Identification phase is meant to detect such an unauthorized attempt. SABAC uses the HSA code generated and image classification process to ensure that the confidence level is within 95-100 percent and similarity thresholds of the system is 1 for the verified users. However, if the values are below 95 percent and 0 for both measures, the system classified the user as not confirmed.

The next phase is the users' data analysis. This is a process which includes tracking of user, validating user action, retrieval of hidden/encrypted data, confirming the transaction date and time stamps, etc. The HSA key is used for tracing who handles a specific activity on the cloud resource. This helps to ensure that the system confidentiality (unauthorized person not allowed into the system), integrity (to ensure data is valid and accurate) and availability (time and date stamp testing to verify that the system is fast enough during authentication and the resource are readily available for use).

The report evaluation phase assesses the process that takes place in the platform. It includes clarifying whether the described and evaluated components are indeed important to the kind of data held by the user. Also, apart from maintain confidentiality, integrity and availability, the HSA key is used for the justification of user's record that was retrieved and analyzed. Information must be:

i.

Identical with the retrieved original.

ii.

The encoded value encrypted is always the same.

iii. The retrieved information is independently checked and analyzed.
iv. If an intruder makes an attempt, a warning message may be sent to an approved person.

Password changes are an important part of this operation, and they should be done on a regular basis. The username, old password, and new password are all provided, with the old password being overwritten. HSA and New image-id are also generated and stored. This approach ensures that the SABAC system's credibility is maintained. The efficacy of the service's accessibility, the evaluation of the user HSA, password modification, unauthorized attempt, timestamp, percentage of trust level, and similarity are all available for reporting. The device also shows the user status; 0 to indicated unauthorized user and 1 to indicate authorized user.

Database Archive is the last phase. The administrator is responsible for the protection and storage of database of the user's data, as well as the organization. For further processing, the HSA and other attributes given are stored in the database archive. The image is used as one of the major prerequisite attribute needed for accessing the database; this is controlled and enforced by encryption of the user attributes and HSA information collected. The database is useful for traceability in the case of any user dispute or if hackers or attackers attempt to unlawfully break into the system. If there is no backup, it means that if a disaster occurs, the company's data is lost. This is one of the shortcomings in some of the existing cloud platforms. SABAC supports the use of a secure backup system that can be used during repairs, downtime, or system failures. To ensure the continuous availability of both service and resources, this is a primary requirement for using this security framework. Figure 2 displays the SABAC Database Archive.

Security metrics (Integrity, Confidentiality, and Availability) were used to assess SABAC model to measure its efficiency. SABAC model performance is enhanced by the HSA algorithm. The integrity, confidentiality, and availability of the data on the IaaS platform are guaranteed by this new access control model. HSA is a symbolically generated code that connects user permission with their attributes, facial appearance inclusive. An encrypted code is generated using the MD5 hashing algorithm using the specified attributes. Each individual has a distinct appearance that defines who they are by nature; this characteristic is converted to a 64-bit hexadecimal value. The names of the user, image, username and password were combined together using message digest to generate one-way encryption code for authentication purposes. The MD5 uses 128-bit memory with minimum of 512-bit message length for generating the hash code. The message length can be smaller than the minimum value required, it is padded by adding a 64-bit value to it.

The Symbolic code is used to create a connection between the picture that has already been captured and the face that was under monitoring. HSA is used to find out who should have access as contained in the access monitoring list and what roles and resources they have access to. For instance, in the SABAC cloud security system, the user can possess individual permission right to do any of the following on the service platform depend on their subscription; right to change, remove, update, and view data.

The HSA uses the MD-5 encryption algorithm to generate the hash-tag code. HSA is a distinctive code that offers a user access to cloud platform services. SABAC is differentiated from other access control models in literature by its symbolic feature. The symbol differentiates one user from another. It's seemingly difficult to alter the user's facial image if other details can be manipulated. Shown in Figure 3 is the HSA process flowchart diagram. Also, the algorithm for HSA is also shown in Figure 4.

There are two stages in the HSA process flow mechanism. These are login and request to use cloud services. In the first stage,

new users are signed up by creating a personal profile. The attributes are concatenated and converted to hexadecimal which serves as input into MD5 encryption algorithm to generate an HSA key. Existing users log-in by providing their username and password, and the device conducts confidentiality authentication and information validation by face recapturing. The second stage is the request for user accessibility to cloud services. The process includes: the request to access the database, the system's integrity authentication check to assess the user's restriction level, the process of face validation, the verification of availability authentication, and the face validation.

The SABAC algorithm is as follows:

```
Begin
    If 'Existing User'
    Input username
    Accept password
    Input HSA
            If 'username' and 'password' is valid
                    Capture face
                    Verify face
            Else
                    Exit
            End if
            If face is valid
                    Generate HSA
                    Retrieve user's attributes
                    Compare user supplied HSA with computed HSA
            Else
                    Exit
            End if
            If HSA is correct
                    Confirm captured face
                    Grant Access
                    Calculate Confidentiality
                    Calculate Integrity
                    Calculate Availability
            Else
                    Exit
            Endif
            While User Is logged in;
                    Capture face at interval
                    Verify face
                    If face is 'not valid'
                            Exit
                    Else
                            Continue
                    Endif
            End While
    Else
            Create username and password
            Capture user facial image
            Convert image to string
            Concatenate (username + password + image)
            Convert code to hexadecimal
            Apply MD5 encryption
            Store attribute in database
            Exit
Endif
```

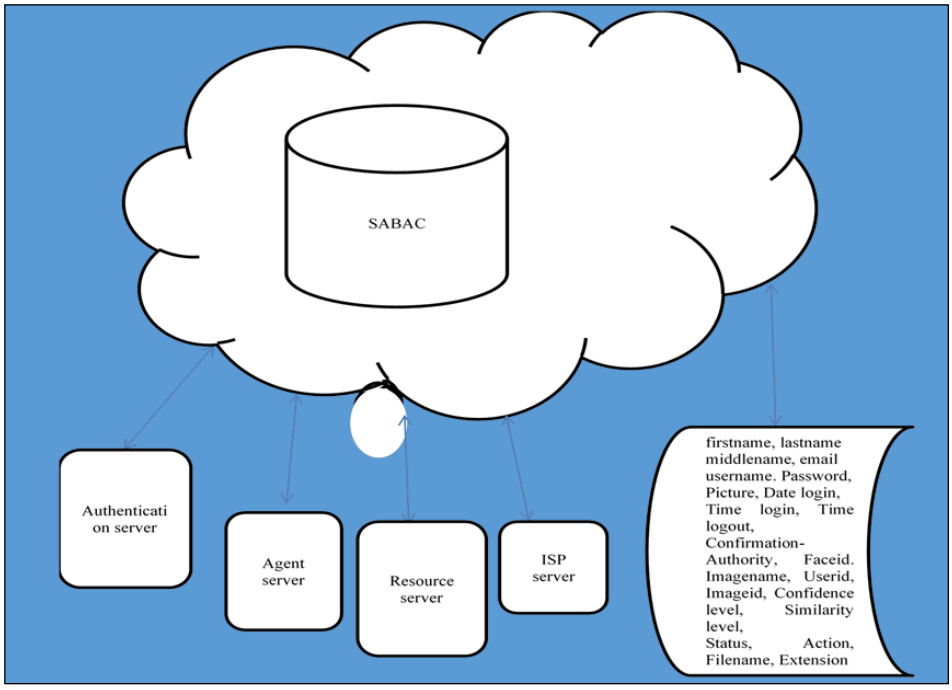

Figure 2: SABAC Database Archive.
Source: Authors, (2021).

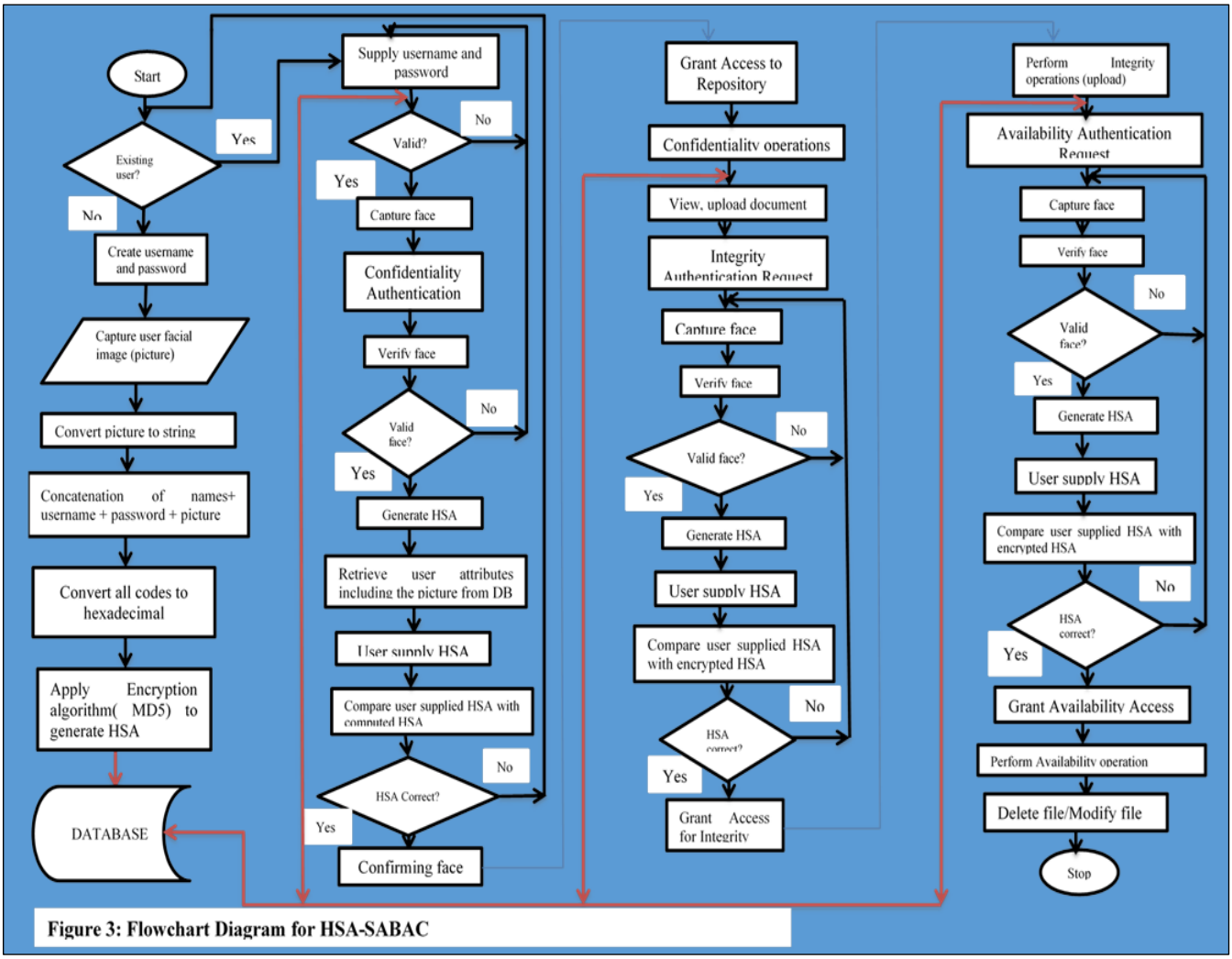

Figure 3: SABAC Model Flowchart
Source: Authors, (2021).

IV. RESULTS AND DISCUSSIONS

SABAC performance is evaluated using three security metrics: confidentiality, integrity, and availability.

i. Availability

This is the number of users authenticated per unit of time which is also called throughput. It is determined in this study by adding the time required to validate a user and the time required to produce HSA. The number of users divided by the time it takes to authenticate each user is the general formula for throughput (availability). The formula for throughput can be deduced as follows:

Let P represent the availability performance, T total time taken to validate user, and U represent the number of users.
The throughput is implemented thus;

Throughput

$$P = \frac{U}{\sum_{i=1}^{m} i\,T} \quad (1)$$

where i=1…m

One way to determine the efficiency of SABAC system is to look at its throughput. In measuring throughput, the length of time the system spent in authenticating each user was recorded. This gives an insight to how fast the system attends to users request and how many users it can handle per time.

Table 1, shows the image validation time and authentication time for each group of users. It is worth noting that image validation time is a subset of the authentication time for each user and it determines significantly the time SABAC spends on authenticating such user.

It is observed from Figures 4, that the image validation time and authentication time is fairly stable while the number of users increased. This makes SABAC to be stable and always available to users as the load increases in the cloud. In figures 5, it is observed that SABAC becomes rugged and maintains an acceptable speed of authentication even as the workload increases.

In Figure 5, the throughput of SABAC was presented. The system become stable as the load increases, for instance, when the number of users is 40 and above, the throughput become higher as the number of users increase. It is observed that the throughput increased by 15,000% to 1,828 users per second between 10 users and 100 users. This implies that the system become more efficient as higher number of users enter the system.

In the cloud security system, the system should disallow unauthorised person from accessing the resource. The system's precision is determined by the following possibilities: false positive, true negative, false negative, and true positive. At all levels of access to cloud services, the system should not allow a false positive.

Table 1: Availability

Number of Users	Image Validation Time (ms)	Authentication time (ms)	Total Time (ms)
10	1.28	4.19	5.47
20	1.29	4.33	5.62
30	1.34	4.41	5.75
40	1.49	4.52	6.01
50	1.55	4.59	6.14
60	1.61	4.7	6.31
70	1.64	4.76	6.40
80	1.66	4.81	6.47
90	1.7	4.87	6.57
100	1.74	4.91	6.65

Source: Authors, (2020).

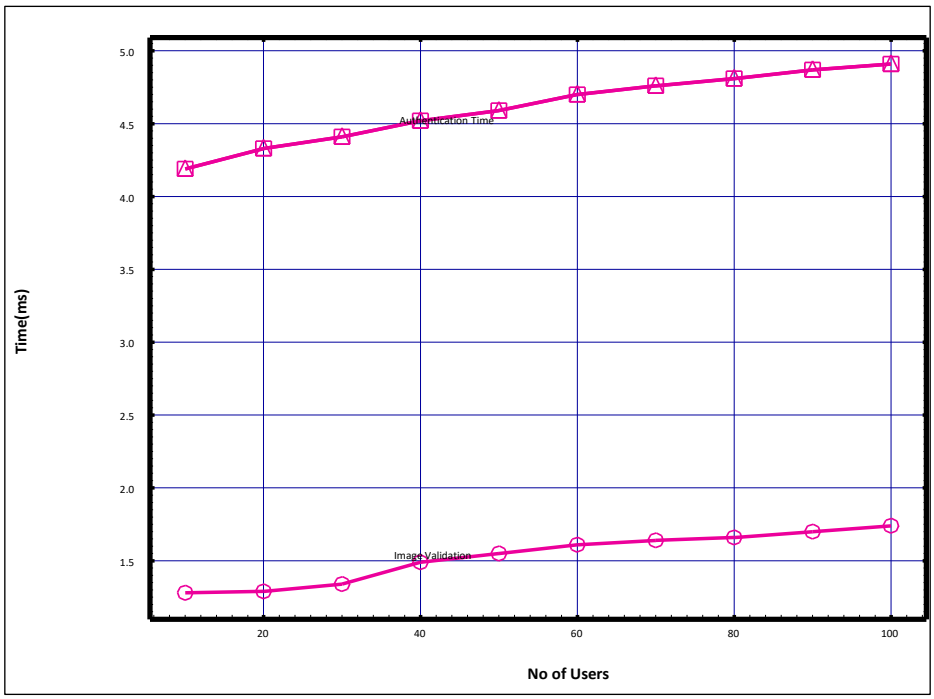

Figure 4: Image Validation and Authentication time.
Source: Authors, (2021).

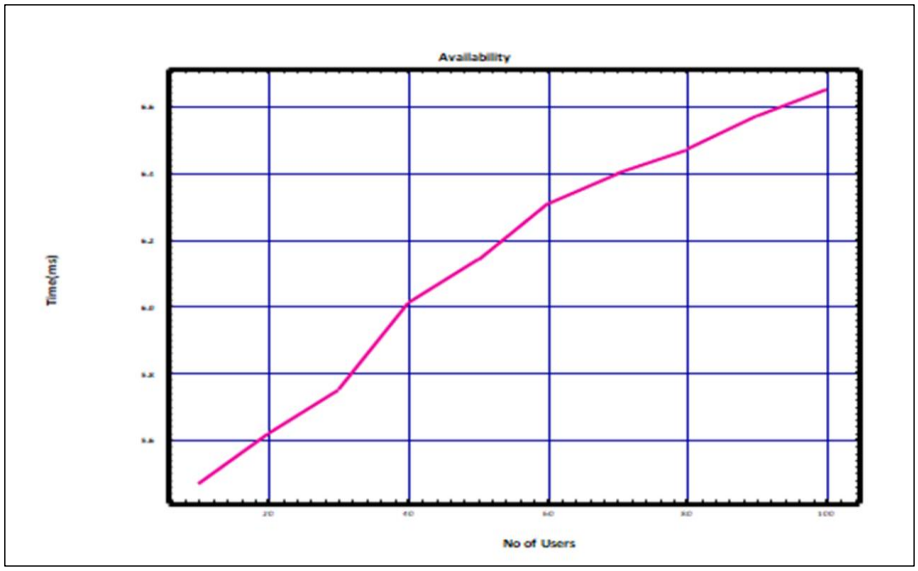

Figure 5: Graph of Availability.
Source: Authors, (2021).

Integrity of the system is tested by checking the similarity value between the symbolic credentials stored in the database and the recent symbolic information collected when the user attempted accessing the system. If there is no discrepancy in the encrypted and the new data, the image is successfully validated. Each successful process is rated based on the criteria incorporated into the classification algorithm. Parts of the determining factors are the pattern of the image, environment and the level of interference in the vicinity where the image is captures.

In Table 2, The similarity threshold for 10 to 100 users are 92.88236 to 99.9980 respectively. The more the number of users in the system the higher the stability and accuracy.

In Figure 6, the integrity graph, the increase in the similarity threshold as more people access the system. The data become stable and the system was not interrupted in any way. The flow of the graph shows that if the number of users increases at 10 folds, there was no case of unauthorized attempt to disrupt the functionality of the system.

Confidentiality is evaluated in this study by the confidence level, which is the percentage of confidence the system has in the user it authenticates. The confidence thresholds is protected using HSA to reconfirm the user privileges to the cloud. SABAC re-evaluates the confidence level of the user at each access point services. The confidence level ranges from 0% to 100%. The higher the rating, the greater the confidence SABAC system has in the user. This authentication system has a minimum trust level of 95%.

Table 2: Integrity.

Number of Users	Similarity Threshold
10	92.88236
20	95.68957
30	95.9957
40	98.49678
50	98.79778
60	98.8965
70	99.8750
80	99.9856
90	99.98975
100	99.9980

Source: Authors, (2020).

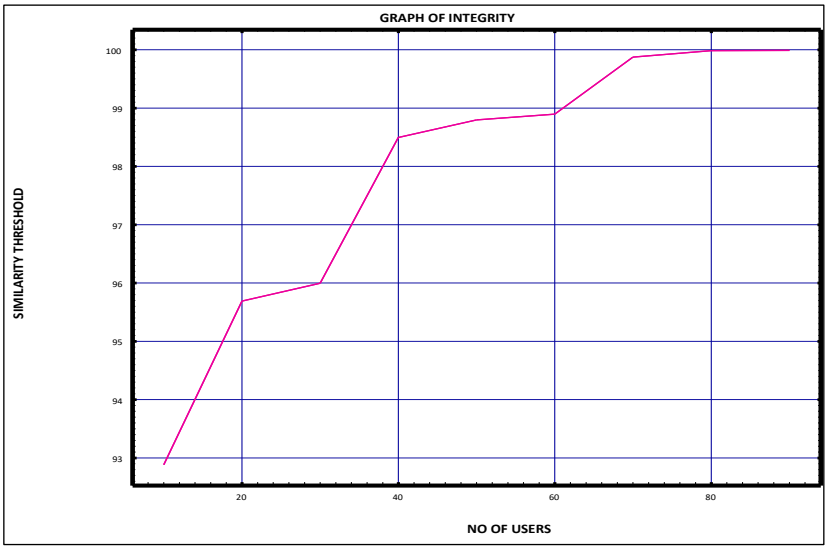

Figure 6: Graph of Integrity.
Source: Authors, (2021).

Table 3: Confidentiality.

Number of Users	Confidence Level
10	99.9999
20	99.9997
30	99.9997
40	99.9995
50	99.9994
60	99.9994
70	99.9994
80	99.9993
90	99.9993
100	99.9993

Source: Authors, (2021).

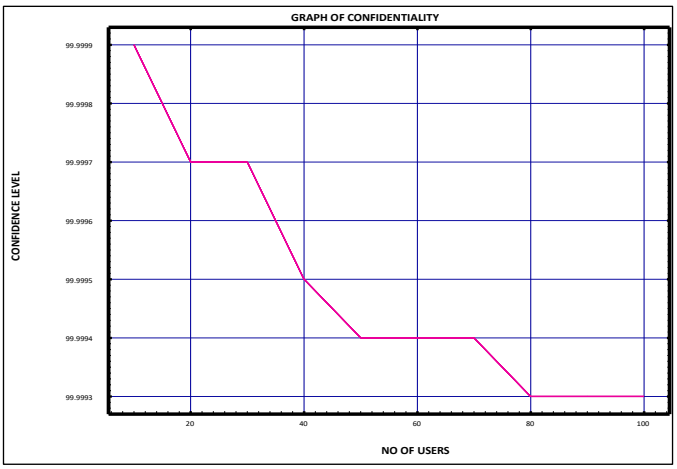

Figure 7: Confidentiality Graph.
Source: Authors, (2021).

The average confidence level for each group is as presented in the Table 3. The presented data shows that the confidence level of SABAC ranges between 99.9993 and 99.9999. This implies that SABAC confidentiality measure is very high and acceptable. It should be observed that as number of users increase, the confidence level tends towards a limit of 99.9990. Figure 7 presents graph for confidentiality.

V. CONCLUSIONS

The value of security cannot be overlooked in cloud computing. It is becoming more and more essential for Internet applications and services. Cloud computing usage is increasing, and it is rapidly becoming main-stream infrastructure. Cloud computing's adoption and use come with major security concerns. As a result, SABAC was built to fix the security problems that come with cloud usage. For successful cloud protection, the developed system was implemented with multifactor attributes fortified with image code. As a result, SABAC is an excellent solution for using cloud services in a safe environment. The use of SABAC as an authentication mechanism would enhance cloud resource security and restore confidence in cloud adoption and use.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Iyabo Felicia Oyeyinka.
Methodology: Iyabo Felicia Oyeyinka and Sunday Idowu.
Investigation: Iyabo Felicia Oyeyinka and Sunday Idowu.
Discussion of results: Iyabo Felicia Oyeyinka and Sunday Idowu and Afolashade Kuyoro.
Writing – Original Draft: Iyabo Felicia Oyeyinka.
Writing – Review and Editing: Iyabo Felicia Oyeyinka and Sunday Idowu.
Resources: Iyabo Felicia Oyeyinka.
Supervision: Sunday Idowu and Afolashade Kuyoro.
Approval of the final text: Iyabo Felicia Oyeyinka and Sunday Idowu and Afolashade Kuyoro.

VII. REFERENCES

[1] Sandeep, B., "Cloud Computing." University Printing House, Cambridge CB2 8BS, United Kingdom One Liberty Plaza, 20th Floor, New York: NY 10006, USA 477 Williamstown Road, Port Melbourne, Vic 3207, Australia 4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi – 110002, India 79 Anson Road, #06–04/06, Singapore 079906, 2017.

[2] RSA, "The Current State of Cybercrime: An Inside Look at the Changing Threat Landscape," 2014.

[3] Teri, S., "Data Insecurity on the Rise." The Verison Business Data Breaches Investigation Report, Financial Management Magazine. www.fm-magazine.com, 2020.

[4] Vincent, C. H., Kuhn, D. R., & Ferraiolo, D. F., "Attribute-Based Access Control, "National Institute of Standards and Technology. CSDL Issue No. 02, pp 85-88, ISSN: 0018-9162, http://doi.ieeecomputersociety.org/10.1109/MC..33, 2015.

[5] Xin, J., "Attribute-Based Access Control Models and Implementation in Cloud Infrastructure as a Service," The University of Texas AT SAN ANTONIO College of Sciences Department of Computer Science, 2014.

[6] Vincent, C. H., Kuhn, D. R., & Ferraiolo, D. F. "Guide to Attribute-Based Access Control (ABAC) Definition and Considerations," NIST Special Publication, 2013.

[7] Mikko, K.,"Enforcing Role-Based Access Control with Attribute-Based Cryptography for Environments with Multi-Level Security Requirements." Aalto University publication series DOCTORAL DISSERTATIONS. Department of Information and Computer Science, Finland, 2016.

[8] Khalid, Z.B., Ram, K., & Ravi, S.," Constraints Specification in Attribute-Based Access Control, Institute for Cyber Security," Department of Computer Science, the University of Texas at San Antonio, 2013.

[9] Richard, K. D. & Edward, C. J.," Adding Attributes to Role-Based Access Control," 2010.

[10] Jin, X., Krishnan, R., & Sandhu, R., "A Unified Attribute-Based Access Control Model Covering DAC, MAC, and RBAC," .In DBSec, 2012.

[11] Sandhu, R. S. & Samarati, P., "Access Control Principle and Practice," Communications Magazine, IEEE, vol 32. no 9, pp 40-48, 1994.

[12] Parminder, S., & Sarpreet, S., "Cross Bread Role-based Access Control for Extended Security At Az-ure in Cloud Computing," International Journal of Application or Innovation in Engineering & Management (IJAIEM): ISSN 2319 - 4847 www.ijaiem.org Email: editor@ijaiem.org, editorijai-em@gmail.com, 2013.

[13] Masound, B., Mohammad, S.K., Soheil, L., & Azizallah, R., "A New Semantic Role-based Access Control Model for Cloud Computing," The Ninth International Conference on Internet and Web Applica-tions and Services. ISBN: 978-1-61208-361-2, 2014

[14] Alshehri, S., "Toward Effective Access Control Using Attributes and Pseudoroles," The Ph.D. Program in Computing & Information Sciences B. Thomas Golisano College of Computing & Information Sciences Rochester Institute of Technology, 2014.

[15] Kim, W., "Cloud Computing Trends: State of the Cloud Survey," 2015.

[16] Huihui, Y., "Cryptographic Enforcement of Attribute-Based Authentication." The University of Agder, Faculty of Engineering and Science, 2016.

[17] Djellali, B., Belarbi, K., Chouarfia, A., & Lorenz, P., "User Authentication Scheme Preserving Anonymity for Ubiquitous Devices. Security and Communication Networks, vol 8, no17, 2016.

[18] Oyeyinka, F. I., Awodele, O., Kuyoro, S. & Oyeyinka, I. K.,. "The Design of Hybrid Cloud Migration Techniques," 2nd National conference and 4th Induction ceremony of Nigeria Women in Information Technology, Abuja, Nigeria, 2016..

[19] Seyed, M. D., & Sara, N., "Authentication Techniques in Cloud Computing: A Review," International Journal of Advances Research in Computer Science and Software Engineering, vol 7, no 1, 2017.

[20] Ako, M. A.,. "Advanced Encryption Standard (AES) algorithm to Encrypt and Decrypt Data," Department of Applied Mathematics and Computer Science, Eastern Mediterranean University-Cyprus, 2017.

[21] Imran, K., "An Introduction to the applications of cloud computing technology in academic libraries," International Journal of Library Management and Services. ISSN:2349-6347; vol 4, no 2, pp 15-24, 2018

[22] Yu, S., Yusuke, N., Noboru, K., & Kazuo, O., "Improved Collision Attacks on MD4 and MD5," 2007.

[23] Rajeev, S., & Geetha, G., "Cryptographic Hash Functions: A Review. School of Computer Science," Lovely Professional University Phagwara, Punjab. International Journal of Computer Science (IJCSI), ISSN: 1694-0814, vol 9, no 2, pp 2, 2012.

[24] Yu, S., & Kazumaro, A., "Finding Pre-images in Full MD5 Faster Than Exhaustive Search," 2009.

[25] Kanickam, L. H. S., & Jayasimman J., "Comparative Analysis of Hash Authentication Algorithms and ECC Based Security Algorithms in Cloud Data," Asian Journal of Computer Science and Technology ISSN: 2249-0701, vol 8, no 1, pp 53-61, 2019.