IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.8, Issue.7, July 2020 E-ISSN: 2347-2693

Big Data Authentication Protocol with Hierarchical Attribute-Based Encryption and Authorization Structure

S.E. Tuase[1*], D. Matthias[2], N.D. Nwiabu[3]

[1,2,3]Dept. of Computer Science, Rivers State University, Port Harcourt, Nigeria

**Corresponding Author: shola.tuase@ust.edu.ng*

DOI: https://doi.org/10.26438/ijcse/v8i7.1931 | Available online at: www.ijcseonline.org

Received: 12/July/2020, Accepted: 20/July/2020, Published: 31/July/2020

Abstract **-** The term big data arose under the explosive increase of global data as a technology that is able to store and process big and varied volumes of data, providing both enterprises and science with deep insights over its clients' experiments. Big Data provides a reliable, fault-tolerant, available and scalable environment to harbor big data distributed management systems thus provide a need store our data at cloud providers place utilizing cloud computing technology. Attribute Based Encryption (ABE) techniques came into existence for securing and providing access control with its many attendant problem, more so with the use of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key-Policy Attribute-Based Encryption (KP-ABE). Big Data is used to maintains and manage valuable data that are store in the cloud. Having the cloud itself not fully trusted possess a lot of issues thereby making the big data in the cloud to face many threats that are not disclosed by services providers. For these reasons we are proposing an authentication protocol for big data with hierarchical attribute-based encryption and authorization structure which will provides a secure authentication protocol for two-level hierarchical attribute-based encryption and authorization structure of cloud big data access control system that will authenticate authorities or users. Our proposed protocol resorts to tree-based signature to significantly improve the security of attribute authorization thus providing data owner a level of security on the data that will require at least two-level of attributes been satisfy before big data can be access. To satisfy big data requirements, we proposed authentication protocol that support two levels of hierarchical attribute-based encryption and authorization structure using a combination of advanced encryption standard (AES), elliptic curve cryptography combining with the hardness of Diffie Hellman theorem (ECDH). Often times, data and file access control encryption were usually implemented with RSA and DSA protocol, which also comes with their own attendant problems, such as computational overhead cost, time sequence for both encryption and decryption key, encryption keys bit length which also culminate in longer period of time for process execution, We proposed a protocol for authentication and key exchange using AES (Advance Encryption Standard) and ECDH (Elliptic Curve Diffie Hellman) that help to resist forgery attack, replay attack, short key bits length thereby enable less utilization of bandwidth and availability on both mobile and desktop computing with robust security based on hardness of Diffie Hellman and Elliptic Curve cryptography algorithm. In addition, we proposed protocol that help preserve entities privacy, our protocol performance is far better than existing protocol, ours enable less power consumption and low bandwidth consumption as its key length invariably has lower bits than other protocol bits lengths. Comparing with the previous studies, we proof and show that our protocol has lower computational and communication overhead. We propose an authentication protocol for big data with the hierarchical attribute authorization structure which require that a trusted root authority grant access to data owner or domain authority that must define data users' attributes and each of the attributes are structure in terms by domain authority. For security of data we use ECDH and AES scheme. Elliptic curve ciphers require less computational power, memory and communication bandwidth giving it a clear edge over a traditional crypto-algorithms. Lately, many companies have adopted the use of ECDH algorithm to improve security efficiency (WhatsApp, Facebook, Firefox, etc.). In hierarchical attribute authorization adopted two levels - Trusted Root Authority (TRA) and Domain Authority (DA). The TRA acts as an authorization manager and DA's in second tier as subordinated to the TRA.

Keywords**:** Big Data, Cloud Computing, Authentication Protocol, Hierarchical Attribute Set Based Encryption, Ciphertext Policy Attribute Base Encryption, Elliptic Curve Cryptography, Diffie Hellman, ECDH, AES, ABE, CP-ABE, RSA, TRA

I. INTRODUCTION

A quick look at the current trend in technology indicates a shift from conventional desktops systems to grid computing and big data. The use of big data to store and acquire valuable information has becomes a valuable phenomenon which is gaining widespread acceptance. Furthermore, a growing number of individuals, corporate, government and non-governmental organization are migrating and exploring the application of cloud computing. Big data (BD) is a word used for datasets involving huge or multifaceted where a conventional processor is unable to handle. BD concerns includes capturing, storing, breakdown, discovery, distribution,

exchange, simulation, application, security and privacy. It is a terminology that describe a voluminous structured, semi-structured and non-structured data that are significantly large and complex to process with an ordinary database and methods, where organizational in most scenarios has volume of data that is very large or that changes very fast or exceeds current processing capacity. An example is Hadoop, which is an open-source software platform developed to save data and run applications on generic hardware clusters. Often, BD provides tremendous capacity to different types of files, with immense processing capacities and facility to handle multitasks or jobs that are virtually boundless. The benefit of big data to organizations, businesses, companies and many large scales and small-scale industries are enormous. In this work we discuss various possible solutions of big data cloud computing security problems and travails. For data protection mechanism, we provided an access control, authentication and authorization with auditing to provide confidentiality and encryption of data. Cryptography offers features which might not compromise data on the cloud thus Big data has become an important phenomenon that is use to collect valuable information, however, the main objectives of Big Data analysis is to handle hefty data to gain substantial details with suitable method of analyzing big data that will be very important in long term which might not be easily achievable using a single computer or server. Advent of big data cloud has provided society with seamless technical support and convenient data management methods which gives owners of data and customers benefit of first-class cloud services through Internet.

More so, security of virtual services is of concerned to big data with entire process been transferred via internet given that end-users entrust their big data to cloud service providers for storage and with business operations which are not entirely dependable. Big data cloud provides massive data storage with massive computing power and ability to deal with virtually limitless concurrent tasks or works for any type of data. In public cloud management, customer interfaces are accessible from internet given that data deleted may not actually be deleted at the cloud end thus posing a security issue in cloud computing which has become serious problems of late. More so, big data is transforming medicine, sciences, engineering, finances, and business. These advances in storage and mining of data technology enable increasing amounts of data to be preserved as analyzed by a change in the characteristics of data held by organizations. Similarly, there is a remarkable form and pace of data processing, and data security is an integral part of an organization's ICT technology architecture. Such standards become even more rigorous as data is migrated to the cloud, and data is made accessible via the internet. Reliable and good authentication with access control systems should therefore be in effect while such tools are deployed. Based on examination of relevant literature and our own research, we will examine general security and privacy criteria for cloud residing records big data. Most companies are using cloud services based on various business models (SaaS, PaaS & IaaS) and four implementation models (public, hybrid, social, and group cloud) at their convenience. There are a number of problems in cloud computing and data protection issues need to be resolved. Such challenges fell into two wider groups, i.e. security issues faced by cloud providers and security problems encountered by customers. All parties are saddle with different responsibilities ranging from encryption, strong password and authentication. Information protection and partitioning of storage segregation needed to be ensured by vendor, thus, provider will be held accountable for data leaks. Many rules and policy have been created by administrator in charge of storage and usage of data; to achieve interoperability of access control various structures came into existence. The outline issue is that this policy work when provider's and clients / users are under a control of domain authorities. To resolve these obvious security challenges attribute base encryption framework of various kinds is adopted, though there is lack of scalability of attributes at multiple levels. The main objectives of many researches on big data are mostly in processing of volumes of data that are require to obtain relevant information. Furthermore, an effective system for handling of big data is very risky, yet it cannot use a single computer or server to handle big data.

Distributing and virtualization have developed with the provision of lots of big data benefits needed services which include distributed computing, virtualization, distributed database, and so on. The progress and benefits that big data cloud provided include varieties of technical supports and suitable data management methods available to desirable consumers, thus, trusting computational capability and limitless storage of cloud services, users will always benefit from efficient services and obtain data from cloud, thus, users will benefit from high-quality services via internet. Recently, virtual storage and data processing services are facilities rendered by cloud service provider which provide clients with insufficient resources to contract out their local big data to cloud service provider's. There are various scheme and model that reveal CSP as not completely trusted and data stored can be retrieved by unauthorize users, which may in-turn subject big data cloud to encounter threats. There is no security analysis readily available to resists forgery and replay attack. Most times, Cloud Service Provider (CSP) usual conceal information from users of a possible security risks in order to continue to gain customer confidence, thus, BD might encounter security threats that includes theft, and unlawful access. Key-policy attribute-based encryption are require to create policy for private key. In spite of different scheme and model that has been developed, design and implemented below are few of the key factors affecting the schemes: Less security in cloud big data. Lot of attacks in the cloud, however, to resolve some of these highlighted factors and problems we are proposing a protocol for users' authentication of big data with hierarchical attribute-based encryption and authorization structure using cryptography framework that

will improve flexibility of user attributes. More so, the hierarchical attribute-based encryption combines ciphertext-policy attribute-set-based encryption with an authorize structure. We are also proposing an authentication protocol that enable two-level structure of attributes and authorization to provide an access control of big data system to authenticate an entity. To achieve this aim, we propose authentication protocol and key exchange protocol that provide multiple levels authentication in hierarchical attribute-based encryption and authorization. Taxonomy of tree base signature enable our proposed protocol to provide security properties for phony attack resistance, repeat attack resistance and concealment / privacy preservation

II. RELATED WORKS

The terms that define architecture of this study, namely Authentication Protocol, Hierarchical Attribute Base Encryption and Authorization structure. Most specifically, Big Data Encryption System and Protocol on Authentication. Limitless cloud computing space, analytics services tempt with Big Data gathering storing capabilities. Key trials include the defense of data security in an untrusted cloud environment, taking an appropriate judgment on data access, and applying access laws. For this reason, researchers have described and documented multiple system models and view management strategies based on cryptographic operations to offer protected and resourceful control of big data cloud access. Cryptographic methods have developed their own standard for lightweight apps, however. Since these strategies allow for the exchange of data between a wide range of users, many accessibility problems still require solutions, especially an Internet of Things (IoT) packages. A good policy measures ought to be adequately descriptive and to include applicable contextual information on user, object, or environment-related properties, attributes or characteristics, representing constantly evolving circumstances and associating with the current environmental behavior in [1] in question. Such high-quality systems allow the data easy to access, good scalability and flexibility feasible for [2]. In distributed system, specifically big data cloud, native cryptographic methods used same key for encryption and decryption operations thus suffered from big delivery with administration problems. [3], in spite of whole advantages, have raised thoughtful apprehensions on security and privacy of cloud server location. Many tools aided digging on farm out of data stored in cloud environment to keep Big data private. Nonetheless, standards cannot regulate permission to designated stored Big Data, or compel permission for policies of [4]. Some applications vary widely, such as e-health, comprises preserving medical records and electronic patient monitoring, law enforcement systems involving data management and troop tracking, decisional based vehicles like traffic automation management and smart cities, sustaining these deployments through the creation of a collusion-resistant network and availability of fine-grained permission control is a major problem, this is because of vulnerability of such application data and the effects of breaching these structures are doing considerable damage to the networks and their users. The most important issue is how to flexibly enter and exit a low-cost computing system (i.e. secure revoking of accounts). A number of works was carried out to create necessary cryptographic methods to overcome the above challenges. To maintain secrecy, [5] introduced the first hierarchical primary assignment framework as regards hierarchical access controls. Specific objects require encryption with a symmetric encryption system, where users are divided into an ordered partially hierarchy which focuses at on assigned privileges to create protection groups. Increasing rank is assigned private and public content to access key of the lower-down levels within the hierarchies. Contrary to the dynamic, broad-scaled design of cloud-based storage, complex changes (e.g. compare, elimination, etc.) are often not required for this process. In fact, a study was undertaken by [6]) to proffer a solution to shortcomings of [7] work by reviewing their primary assignment techniques that efficiently solves main storage issue as well as complicated improvements such as account revocation or class revoking.

However, all adjustment that happens within reinitialize the criteria of modifying classes and required reduced hierarchical classes. The laws governing relations are not sufficiently descriptive such systems also depend on the assumption by RSA that a small key size of 1024 bits will be used to alleviate discrete logarithm issue. For that reasons, high computational cost is needed for key generation and key derivation processes, hence, these devices are not appropriate for enterprise applications. [8] recently introduced hierarchical primary assignment. A similar structure uses a sole secret and few public information to retrieve decoding key, these does not require the release of public information in lower class key derivation systems, which are based on the principle of access control of chain partitioning. Additional key is to be transmitted to designated user with multiple secret keys but not more than number of partitions, providing an improvement in number of data submitted through hidden channel. This does not involve specific modifications viz omission, redesign, and procedures, though, improvement is accomplished by enforcing attribute-based access control, that grant users access depending on qualifications.

[9] scheme discussed the features of ciphertext attribute base encryption with key-policy base attribute encryption and key distinction lies on how to attach secret key to access control system with specific features of an entity. A key policy attribute base encryption uses secret key to associate an access policy and set of attributes where key is encrypted with the data. Alternatively, separate access policy is allocated to every authenticated data object in CPABE and attribute sets is assigned a secret key for data decryption. The result of inserting access policy into secret key in KP-ABE allow system to select set of attributes but are denied rights to determine the user that can access their

ciphertext. It would be responsibility of attribute authority to decipher ciphertext by generating decryption key for approved users, [10]. Nevertheless, given CPABE merits, major problems are not resolved by [11] and [12] thus, current CPABE programs have the revocation and preservation of varieties of attributes. In term of revocation, domain is critical to any access management scheme to simulate the complexities of storage in cloud. To these ends, several researchers had dedicated significant attention to extending the existing systems and addressing the question of removal with small computing cost to be applicable to IoT applications. It should be remembered that much of new CP-ABE work achieved data security, protection, fine-grained access control and concise system. [13] and [14] to be able to assess the existing work critically on the basis of the above described difficulties and conditions. However, the challenge of carrying out a good revocation process still remains open. Consequently, in this report, we focus mainly on conditions that yet to be effectively met, such as revocation and resistance to plot. Goyal's work is the first to examine the revocation procedures used in single and multiauthority schemes based on several common variant of ABE and CP-ABE. Virtually all the existing papers discuss specific types of access policies that appear in the available literature (e.g., monotonous or non-monotonic access policies) thus leaving the problem of revocation unsolved. Assigning an access policy with ciphertext implies ciphertext decides and determine key that is capable of decrypting the plaintext, giving owners' rights over its outsourced data [15]. A qualified user with right access policy permission will decode the encrypted data, thus, advantage of CP-ABE permit data to be store in an untrusted server without big data entry protection checks. [16].

[17] proposed an alternative solution to ensuring safe data sharing by splitting big data into small pieces. There, broad data of client was used, which did not require any additional security requirements before being stored in the cloud. Cryptographic virtual mapping was also used to ensure the security and confidentiality of big data. Here, the effectiveness of this mechanism was assessed on the basis of the overhead and effectiveness of the proposed scheme. However, this study required an increase in the level of security before the data were stored in the cloud. [18] developed a Remote Data Auditing (RDA) system to ensure the protection and integrity of data stored in the cloud. The Divide and Conquer Table (DCT) was used here to effectively endorse data operations such as adding, appending, modifying and deleting. The main aim of this study was to reduce the required cost of commemoration and computation with increased efficiency. The key weakness found in this analysis was an increase in overhead memory in the cloud storage system. [19] has implemented a new multi-share system to protect the privacy of data stored in the cloud. In this study, the benefits of proxy re-encryption were integrated with an anonymous data security technique.

III. METHODOLOGY

Focus of this section is to provide and model a framework for authentication protocol of Big Data with hierarchical attribute base encryption and authorization structure using a bilinear pairing technique which is base on which many cryptographies are model and elliptic curve cryptography technique to secure big data access and storage. This research will provide an improve security system to save important data from attackers by preventing collusion attack and revocation attack. For this; am going to compare previous security algorithms and try to remove their drawbacks and improve security of big data in the Cloud. In this research will provide an effective cryptography algorithm with a combination of AES and ECDH cryptography technique which are used for data encryption and key generation for a secure and efficient system that will eliminate the problem of an existing systems. The methodology that is adopted in carry out this research is qualitative which like other previous research that make use of various methodology viz; theoretical, simulation, explorative, incremental and scientific though an analytical research was used at the beginning to explore the strength and weakness of previous work done.

Systems Architecture: To administer distributed data files, data users will need to login to the system using key exchange shared by the data owner who is an authority to access an encrypted data of interest from big data cloud and decipher them to plaintext. Figure 1 and 2 is a proposed system architecture showing the access level of the entities and attributes. Each data owner/user is administered by domain authority which are controlled and managed by trusted authority.

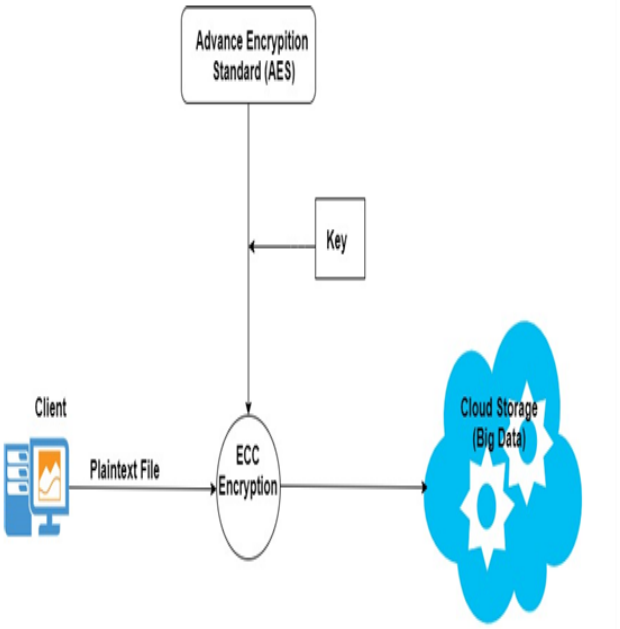

Figure 1: System Architecture Component (Authentication)

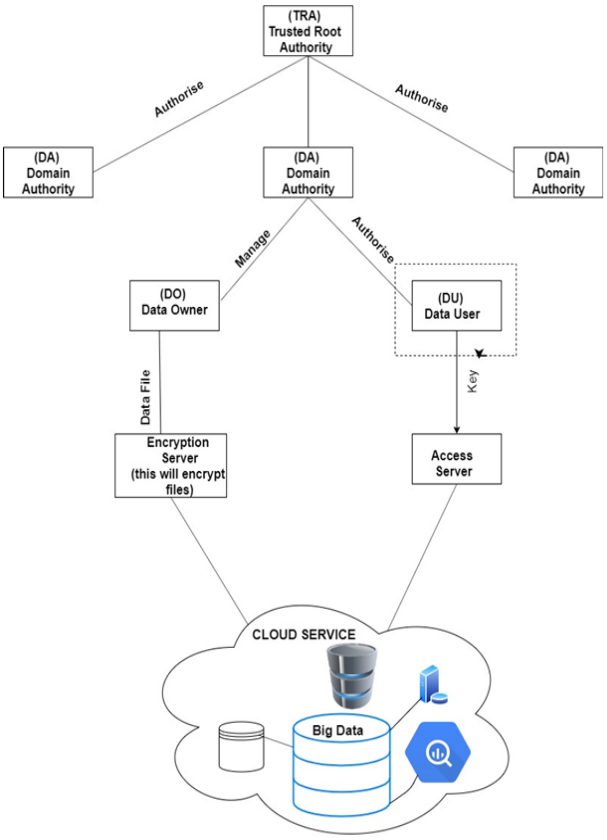

Figure 2: Proposed System Architecture

User Authentication : Majority of big data components implement permission control based on user or group thus if a user is not used, the service just trusts the identity information provided by clients, that is, so long as a client accesses the services using a user ID, the service uses the user for permission check. This issues mostly results in counterfeiting other users. A secure user authentication that is enabled is implemented by verifying user identity information before granting it access.

Let p be a large prime and E be an elliptic curve over F_p given by equation 3.10 of Weierstrass

$$y^2 = x^3 + Ax + B \quad (1)$$

for A, B ∈ F_p. Given that the curve is not singular, then we have $4A^3 + 27B^2 \not\equiv 0 \pmod p$. We define a way to count the rational points on E, that is, a process for calculating number of point on E with x, y Fp{ x, y) on E. Most of what we say applies to elliptic curves over any finite base field. Let E(F_p) denote the set of rational points of E, its easy to see that the number of points in E(F_p) with given X-coordinate x ∈ F_p is 0, 1 or 2. More precisely, there are

$$1 + \left(\frac{x^3+Ax+B}{p}\right) \quad (2)$$

rational points on E with X-coordinate equal to x. Here $(\frac{\cdot}{p})$ denotes the quadratic residue symbol which include the point at infinity, and set of rational points E(F_p) of E thus has cardinality

$$1 + \sum_{x \in F_p} \left(1 + \left(\frac{x^3+Ax+B}{p}\right)\right) = 1 + p + \sum_{x \in F_p} \left(\frac{x^3+Ax+B}{p}\right) \quad (3)$$

This implies that evaluating the sum

$$\sum_{x \in F_p} \left(\frac{x^3+Ax+B}{p}\right) \quad (4)$$

Hierarchical Attribute Set Based Encryption (HASBE):

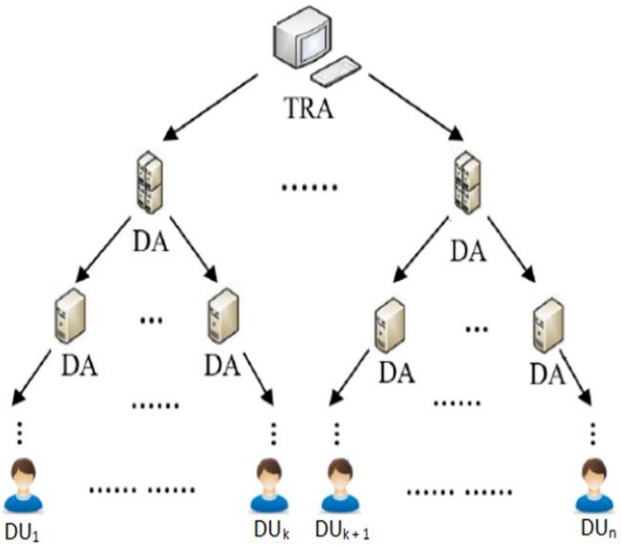

Figure 3: Hierarchical Tree structure

Wang et al (2017) derive from this scheme the Hierarchical Attribute-Based Encryption (HABE). This is for fine grain control of networking of cloud computing applications. It is a blend of HIBE and CP-ABE. Our propose research contains four categories of entities: N Attribute Authorities (denominated as DA), Cloud Server, Data Owners and Data Consumers. Domain Authorities have analytical capabilities, provided that certain attributes partially include user-identifiable personal information. When applying DNF, multiple N disjoint sets are combined to form a complete authority and each N disjoint set is regulated by each domain authority. Therefore, only part of the attributes are known to each DA. The Data Manager outsources encrypted data files to the Big Data Cloud. The Cloud Server has an ample storage capacity and does little but store massive files. New Data Users can request private keys from all DAs that do not know which attributes are regulated by which DA's. When data users ask the authorities for their private keys, DA produces a composite private key and sends it to users. All computer users can access any of the encrypted data files if their private keys satisfy the TRA privilege tree and can perform a privilege pr-related operation. The server performs an activity pr if the user's credential is authenticated via the TRA or DA tree of privileges. Therefore, in the HABE scheme there are several keys with different uses, first we give a description of the most relevant keys to use as a reference, then the HABE scheme is represented by adding the following randomized polynomial time algorithms:

Setup Mk, Pk: This input algorithm is just a security

parameter. Attributes based authorities are running this algorithm to calculate the system-wide public parameter Mk as well as the authority complete public parameter yes, and to calculate the master key Pk individually. Key generated (M, Pk, U) by MU: this algorithm enables the user to interact with each attribute authority and obtains a private key MU corresponding to the input attribute set of Mu.

Encrypt (M, N, {TRA}pr value {0, ... ,j−1}) (CPT, Ve): This algorithm uses the public key M, the message N, and the privilege tree collection {TPA}p value {0, ... ,j−1}, where j is encrypted. Encrypts the N message and returns the CPT ciphertext and the Ve verification set so that a user can perform a particular cipher text operation if and only if his attributes match the corresponding TRA privilege tree. As specified, L0 is the privilege to read the file.

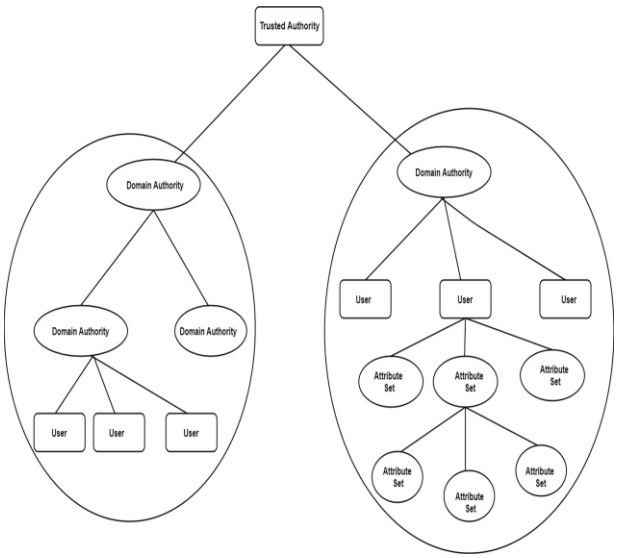

Figure 4: Access Tree Structure

Decryption (AK, MU, CPT) by-N or authentication parameter: this algorithm will be used when managing a file (e.g. reading, alteration, deletion). This takes as input the public key Pk, the ciphertext CPT, and the private key MU, which has a set of attributes Au and corresponds to the GIDu of its holder. If the set Au satisfies any tree in the package {TRA}pr When the authentication parameter is successfully checked by the Cloud Servers that use Ve to validate it, the request for an operation will be processed.

- Phase1: The framework not only provides the privacy of data information, but also requires the privacy of identity by decentralizing the central authority to mask the identity of origin and thus achieving semi-anonymity. Subsequently, the algorithm fully masks an identity that helps to maintain absolute anonymity.
- Phase 2: Program uses the AES (Attribute Encryption Standard) algorithm. The algorithm is used to secure classified information and is used around the world to encrypt and decrypt sensitive data. AES is made up of three block ciphers. AES-128, AES-192, AES-256 and this each cipher uses 128 bit blocks using 128,192 and 256 bit cryptographic keys to encrypt and decrypt sensitive data. The ciphers built in this algorithm use the same secret key for encrypting and decrypting. The various rounds of keys that are executed. Increasing sequence consists of steps that include substituting, transposing and mixing plain text. The plain text is then converted into a cipher text.
- Phase 3: There are four types of systems: N Attribute Authorities (denominated as DA), Cloud Server, Data Owners and Data Consumers. The client may be a Data Owner and a Data User in a single session. Data owner encrypts and uploads files to a massive cloud storage server. Data Client decrypts and installs files from a massive cloud storage server.
- Phase 4: In order to perform certain file operations and to have unrestricted access to that information, the data owner and the data user should participate in the application. Once they are registered with a time password, a specific key will be sent to their registered mail address.
- Phase 5: Upload and open files by the user. User data owner / user demands permission from the domain authority. The authority shall provide the data owner with a public key and the user with a private key. Issuing authority keys and authentication in our program succeeds with the use of attribute-based encryption.
- Phase 6: Ciphertext Policy Attribute-based encryption is a form of public key encryption that depends on the secret user key and the cipher text. Under such a method, the decryption of a ciphertext is only possible if the set of attributes of the user key matches the attributes of the ciphertext. Collusion-resistance is the essential security component of CP-ABE. An adversary possessing several keys will be able to access data if at least one single key provides access to the data.
- Phase 7: The keys issued to clients by the domain authority (data owner and data user) can be used to perform operations and to access files in and out of the Big Data Cloud. HABE uses a normal disjunctive form policy which assumes all attributes that are administered in a single conjunctive clause by the same domain maître. So does multiple domain masters can administer the same attributes in accordance with specific policies that are most complicated to implement in practice, and can have problems with multiple value assignments.

Trusted Root Authority Algorithm to authenticate its Users

TRA:

- generates G → G_0, V_{11j}→Z_p, y→G_1;
- sends (G, V_{11j}, y) to the j^{th}user.

j^{th}user:

- generates m, I_{11j}→Zp;
- computes f_{11}j, y_{11j}, z, $h_{11j;}$

- sends (f_{11j}, y_{11j} , z, h_{11j}) to the TRA.

TRA:

- computes u1.

if $1 \leq i \leq n$ then
if ê (f_{11j} , y_{11j}) . z^{-h11j}= u_{11j} then
the j^{th} user is legal.
end if
end if
end if
end if

User E_{11j} then calculates a signature as $f_{11j} = G_{11j}(v11j+ h_{11j})$. User E_{11j} then calculates for authentication an assist parameter Z= ê(G, y). User E11j will need to transmit (f11j, y11j, z, h11j) to the TRA in a secure channel after extracting the keys and security parameters.

Verify: After the TRA receives user E11j security parameters, it verifies the validity of E_{11j}, so verify process is summarized as follows:

1. The TRA computes an authentication value u_{11j} = ê(G, y)^{v11j} according to its security parameters. Note that these security parameters have been generated in the KeyGen phase.

The TRA checks whether the condition ê(f^{11j}, y^{11j}) . z^{-h11j} = $u1_{ij}$ is satisfied. If it is satisfied, the TRA can be sure that user E^{11j} is an authenticated user in this round.

In figures 2a and 2b, the components enclosed in dotted lines are indicated: one involves calculations and the other involves data processing. The fundamental activity in the first one involves scalar multiplication. The subsequent one carefully executes the KDF module (which produces the keys), E and MAC. Depending on the operation performed, whether encryption or decryption, the MAC data source is different.

Security enhancements: A robust two-factor user authentication scheme with public-key algorithm is invaluable. As a matter of fact, this idea has been generally adopted by large new authentication schemes while the main challenge lies in correctly implementing the public key algorithm. As a result, we'll show the subtleties of implementing a public key algorithm in depth. Notice that because of its high efficiency the elliptic curve cryptosystem has been commonly used in authentication schemes. The elliptic curve algorithm is then deployed by our system to hit our secure authentication system. In this case, our scheme implements an initialization method for parameters to evaluate the parameters of the elliptic curve cryptosystem, similar to the Amin et al scheme.

1. Add public-key algorithm to prevent guessing and assaulting forgery. We've shown the crucial point in responding to such an attack to add a complex non-public parameter in Gi. Here we present the ideas of how to achieve that the reciprocal Di parameter between RA and Ui is a function extracted from (IDi, PWi) for Ui. And any parameter depending on the transformation of (IDi, PWi) is meaningless, as a parameter of this kind equals Di. Therefore, with the help of the public-key algorithm, Ui and RA will share a new parameter in each session. Ui will essentially select a random number Ni, computeX1= NiPub, X2= NiP where Pub= psi, then let X1 be the hidden dynamic parameter in Gi. Notice that calculating X1 is easy for Ui and RA, but for A there is another unknown parameter in Gi that prevents A from executing the dictionary attack II offline.
2. Application of public-key algorithm to provide user anonymity. In Fi we cover the identity-related PIDi parameter as PIDi for h(X1) using X1. Since Fi is updated with Ni it is anonymous to the user. Note that X1 is the complex parameter we apply to under "user anonymity."
3. Using the public-key algorithm to obtain anonymity and maintain confidentiality forwards, we set the ECDH problem in the session key as follows: let SK consist of X3 / Y3, where X3 / Y3= NiY2= NjX2,Y2= NjP. As a consequence, computing SK is equivalent to solving the ECDH problem, which can not be solved in the polynomial time, for the one (including RA) who does not know Ni / Nj. Consequently, the secrecy of forwarding is maintained.

Improvements in Efficiency: Note that previous schemes need only a certain hash operation, by reducing redundant parameters or calculations as follows, we optimize certain output aspects of the scheme:

1. Reduce the number of random numbers selected by Ui to one during the User Registration process. In Amin et al. scheme there are two random numbers since they can actually originate from each other, but the "energy" of computing them is the same, meaning they are "equivalent." Accordingly, it is necessary to use one random number, which saves storage space and computational power.
2. Reduce hidden key number in RA. The hidden main y has no impact on safety improvements. It also makes the CSj register move and CSj authentication more complicated. So we set only one device secret parameter and simplify CSj's register step. Those changes also include other computational efficiency enhancements.

Registration: RA selects two large primes $\{p, q\}$ and a medium integer $n0$ ($24 \leq n0 \leq 28$). Let Fp be a finite field, (Fp) be an elliptic curve over Fp, and G be a q-order subgroup of E(Fp), then RA chooses a point P in G and a long-term secret key $x \in Z*$p and computes its public key Pub as xP. In Big Data environment, the cloud server and the user registration phases are conducted as follows.

For the Cloud Server CSj

1. $CSj \Rightarrow RA$: $\{SIDj\}$.
2. $RA \Rightarrow CSj$: $Skeyj = h(SIDj \| x)$.
3. CSjstores $Skeyj$as a secret key.

To the UserUi

1. Ui {Ai, PIDi}: A new user Ui first selects the PWi password, IDi name, and bi random number as his / her personal information, then calculates the

registration parameters as follows: Ai = h(PWi = bi), PIDi = h(IDi = bi), and initiates the registration process by sending the {Ai, PIDi} request to RA.

2. RA [Ui]: {Ci, Ei, Pub, n0,h)}. RA will first verify whether the PIDi was used via the user-list traverse to ensure the validity of the user accounts. If open, RA selects a single random number ri for Ui and calculates Di = h(PIDi = x ri), Ci = h(Ai = PIDi = Di) mod n0, Ei = Di = Ai, then enter Ui specific parameters{PIDi, ri, Honey List=0} in User − list. Please notice the number of authentication vulnerabilities listed in the Honey List. RA finally acknowledges Ui 's registration by supplying an I d with {Ci, Ei, Pub, n0,h)}.
3. Ui enters DP into the card in which DP = h(IDi [PWi]) = bi.

Login phase: When the user wants to use a cloud account, as shown in Figure 3.19, Ui will use the details to prove his / her authority when logging in.

To test Ui's validity;

1. Initiate request for entry.
2. Ui [CSj]: {Gi, Fi, Zi, X2} login order. Ui enters {ID, PWii}, then we calculate bi = DP = h(IDi = PWi), Ai = h(PWi = b), PIDi = h(IDi = b), Di = Ei = Ei = Ai, Ci = h(Ai = PIDi = Di = Mod n0. To verify the valid Ui, the system compares C Ui with the stored Ci. If you have C ubiquitous Ci, exit the session.

Otherwise, the card accepts the validity of Ui and initiates a request for access to Ui: pick a random number Ni [1, q − 1), measure $X_1 = N_i$Pub, X_2 = NiP, G_i = (PIDj = SIDj − X1).

Basically, it transmits {G_i, F_i, Z_i, X_2} to CS_j, and eventually, it transmits {G_i, F_i, Z_i, X_2} to CSj.

Authentication: Upon request for entry, the CSj must do some calculation to add its specific parameters and forward the request to RA because it is unable to verify the request 's validity. RA will then test the user 's validity and the cloud server, respectively, and help them negotiate the session key. The entire steps towards authentication are as follows.

1. {Y_2, K_i, PSIDj, G_i, F_i, Z_i, X_2}. CS_j selects a random numberNj, calculates Y1 = NjPub, Y2 = NjP, PIDj = SIDj = Y_1, K_i = h(Y_1 = Skeyj = Gi x2 = Zi), then sends it to RA {Y_2, K_i, PSIDj, G_i, F_i, Z_i, X_2}.
2. R.A. {Vcs, Qcs}. First, RA will test U_i by calculating the predefined shared parameter Di and the possible shared secret parameters X_1: $X'_1 = xX_2$, PID'i = Fi = h(X'_1), D'i = h(PID'i x x ri) where ri is retrieved from the user – list by PID'i, SID j = Zi h(D'i X_2). If Ui's reaches a predetermined safe vaslue or the obtained Gi ib (PID'1 ibid. SID"j ibid. X'_1 ibid.), RA treats Ui as an opponent and if it reaches the predetermined value, RA suspends the card before Ui registers, instead refuses the application. If not, RA will continue to verify the cloud service authenticity as follows: measure Y"1 = xY2, SID"j = PSIDj = Y1. If SID"j , exit (RA thinks CSj is not the server that Ui actually wants to access); otherwise continue to calculate Skey'j = h(SID'j = x), K"1 = h(Y"j In the end, RA tests CSj's authenticity by checking if K"i bb"j Ki. If they are not equal, CSj will not pass the authentication from RA, then the session will be terminated. Otherwise, RA must authenticate CSj, then help them to set the session key as follows:

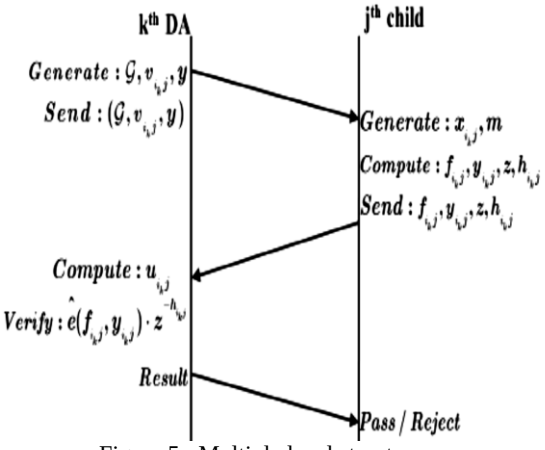

Figure 5 : Multiple level structure

3. Computes Vcs = h(Y"1 qcs} to CSj. Note that Vcs and Qcs are used for CSj and Ui, respectively. Computes Vcs = h(Y"1 qcs) to CSj. They are used to check RA validity, and then inform CSj / Ui that Ui / CSj is actually producing X2 / Y2.
4. Ui CSj {Y2, Qcs}. CSj calculates Vbcs = h(Y1 some Y2 somewhat like Skeyj [X2]), then checks if Vbccs somewhere like Vcs. If the equation is not satisfied, CSj refuses an application from Ui. Otherwise, CSj assumes that Ui is a valid user generating X2, then CSj calculates the session key as SKj = h(Y2 somewhere in X2 somewhere in Y3) where Y3 = NjX2. CSj finally transmits {Y2, Qcs} to Ui.
5. Upon receiving this post, Ui will first test RA 's validity by comparing h(X1 ubiquitous X2 ubiquitous D ubiquitous Y2) to the Qcs obtained. Ui trusts RA and CSj if they are equivalent, then he / she calculates their mutual session key as X3 = Ni = Y2 (= Ni premature (Nj premise P) = Y3), SKi = h(Y2 premise X2 premise X3). The authentication process is completing successfully until now.

Replay Attack: To prevent replay attack, we avoid replay attack via the random numbers. We use one of the message flows {Gi, Fi, Zi, X2} as an example to illustrate: assume A eavesdrops {Gi, Fi, Zi, X2}, then replay it with CSj. Since A does not know Ni, he / she is unable to calculate the correct session key although the replayed message can pass RA verification. Therefore, A derives no benefit from such an attack. Equally, it makes no sense for A to replay other message flows. Consequently, our scheme is free from replay attack.

User Impersonation Attack: As discussed above, A

cannot speculate on IDi and PWi, nor can it replay {Gi, Fi, Zi, X2} to impersonate Ui, let alone the opponent without an ID. Therefore, there is only one feasible form left: construction of {Ga, Fa, Za, X2a}. To construct this post, a selects Na, computes {X1a, X2a}, forges PIDa and Da, calculates {Ga, Fa, Za, X2a}, and finally sends it to RA. Nevertheless, RA will fail to locate such a PIDa in User − list or calculate a Da equal to Da after RA gets C2a, and PIDa, all of which lead to failure of Ui authentication. RA states that A is not a legal user and thus the attack fails.

Server Impersonation Attack: According to our review above, A is unable to perform the replay attack in order to impersonate CSj / RA or find ways to measure Skeyj / x, so A cannot impersonate CSj / RA.

IV. RESULTS & DISCUSSION

Data capacity is either equal to the number of system attributes, or relative to the size of the tree accessed in the layout of the customer view. This research offers a scheme which prevents the customer from accessing information from the big data server / provider. The data owner uploads data to the registry repository, and the owner encrypts the file and stores it for storage purposes in the cloud; the owner is entitled to modify the data file policies by changing the attributes as they deem fit. When the user has the right to access the file, the data user / user can only use the encryption key to open the data files. The domain authority provides all the privilege at any user level, and the data users are governed by the domain authority. Registry service provider maintains a cloud to provide storage facilities for applications, data owners encrypt their data files and store them on server to share with web users. In order to access the shared data files, computer users upload encrypted web resources. Authorizing individual is responsible for creating and transmitting device parameters and root master keys, and authorizing high-level domain authority. Domain authority is responsible for delegating keys to national authorities, or users in their territory, at the next level. Data users download encrypted data files of their choosing from the cloud to view, and then decrypt, the shared data files. Each data owner / consumer is subject to a jurisdictional authority. The authority of the parent jurisdiction, or the trusted authority, controls a domain power. In order to access the shared data files, data consumers download encrypted data files of their interest from the cloud and decrypt them. Any data owner / client is under the jurisdiction of a domain authority. A domain authority shall be governed by its parent domain authority, or the trusted authority. In order to access the shared data files, data consumers download encrypted data files of their interest from the cloud and decrypt them. A domain authority regulates any owner / consumer of the data. A domain authority shall be governed by its parent domain authority, or the trusted authority.

Evaluation of Results: Below are some of the security principle and unique proofs associated with our protocol. Our proposed authentication protocol stably prevents the forgery attack. Our protocol may endure the assault against the forgery with the bilinear paring in the authentication protocol. We check it to show the safety, based on the collision-resistant attribute. In the authentication method of our protocol, one party cannot use preceding safety parameters to pass authentication from its parent. This ensures the new authentication protocol effectively survives the replay attack. We identify as an entity in a round of distribution of attributes that is authenticated by its parent. During this round, the entity must pass the authentication over again. It has not received authentication requirements from its parent, however. Then Entity Elliptic must use the previous security requirements to seek to pass its parent's authentication. Identity information is hidden in our protocol's hash function, so if an entity wants to pass its parent's authentication, it needs to randomly pick one m for the collision resistant Hm. The hash function is a one-way feature that is not reversible and the method can protect the entity's privacy as its parent verifies its validity. The protocol can protect the identity of the users in authentication process, which means it can provide the privacy security properties. Privacy protection is also a basic security necessity that demands that user or entity identity details be protected, which is of the utmost importance when developing an authentication protocol.

Table 1: Comparison of computational and communication cost

Scheme	Computational cost	Communication cost
HASBE	26E.+3Add.+2H.+3Mul.+1P	8lG + 8lZp
Our Protocol	3E.+1Add.+2H.+1Mul.+2P.	4lG + 4lZp

E. : Exponentiation; Add. : Addition; H. : Hash Function;
Mul. : Multiplication; P. : Bilinear paring;
lG : Bit-length of parameter in G;
lZp : Bit-length of the parameter in Zp.

Theoretical Analysis: To show the usefulness of our protocol, we are equating the overhead numerical alignment of our authentication protocol with that of the HASBE. Supposing E. Is, Add., Mul, H. Therefore P. is operations of exponentiation, addition, hash, multiplication, and bilinear paring. On the one hand we compare the HASBE and our Protocol in terms of computational cost. We count the number of operations in key and security parameter generation phases of the prior scheme. We will note from TABLE 1 that the computational costs of HASBE are 26E.+3Add.+2H.+3Mul.+1P .. Because it mainly relies on random masking techniques, and very large quantities of operations. Notice that the tree-based signature is used to design the authentication mechanism for our protocol that is attributable to tree-based signature functionality, our framework has less computational costs. From TABLE 1, we can see that the computational cost of our protocol is only 3E+1Add.+2H.+1Mul.+2P .. The computational costs of our protocol compared to the existing HASBE are significantly reduced. Particularly in our protocol the costs of exponentiation operations are

reduced by eight times compared to the HASBE costs. Conversely, we equate the communication costs between the current HASBE and our protocol. The lG and lZp represent the size of group and finite field parameters. We'll infer from TABLE 1, that the contact expense of the HASBE is 8lG + 8lZp. Owing to decentralization authentication, our protocol has lower communication costs. We can also see that the coordination cost of our protocol is 4lG + 4lZp, relative to the current HASBE, the communication cost of our protocol is greatly reduced. Lastly, we can conclude that our protocol is more cost-effective in terms of technique and communication than current HASBE.

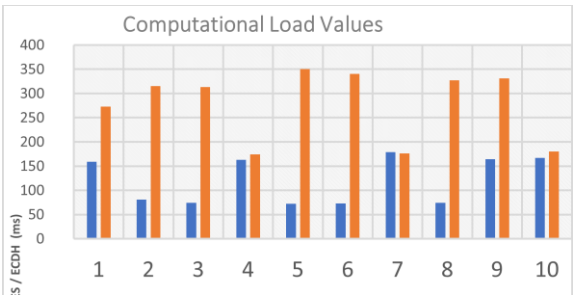

Figure 6:Computation load values

Throughout the authentication method, our protocol can protect the identity of individuals, which means our protocol can provide privacy protection rights.

Preservation of privacy is a basic security prerequisite in security cultures. This requires that when developing a security assurance system, the protection of the identity information of users or organizations must be considered. A comparison of our proposed protocol with related work aimed at evaluating how well the efficiency of the authentication protocol with the key distribution performs better.

Computational overhead: Under our proposed program, the TRA and customer's big data are on cloud service provider, so the time needed for the authentication process and data protection is less compared with previous schemes. In previous schemes, third party auditor and data had been on various pages, and obviously has more space in the authentication method of the previous scheme.

Authentication of Data Security: In our proposed scheme the authentication module plays a role for the intermediates. Without it, neither the cloud service provider can access the authentication information, nor the client owner can.

Name	Profession	FileName	Signature	ID	Permission	value
aa	Doctor	paper titles.txt	Network	1234	Reject	xmXTC7XDqM23zef8yIS9Sw==
aa	Doctor	paper titles.txt	Network	6002	Upload	xmXTC7XDqM23zef8yIS9Sw==
Maha	Staff	image.txt	secret	1159	Upload	3CtO9722T3V+ydH0IatvzA==
Sakthi	Doctor	paper titles.txt	Block	4314	Upload	c0xY05afPhE3gnXL97dODw==
tuase	Select Profession	Rsu.txt	1234	1234	Reject	gcGIKbMxgEstxp2HXXIZyQ==
tuase	staff	setup Algor code.txt	4321	1234	NULL	t68G+LcVHkLy+YVHgz/cMg==
tuase	Staff	pseudo-random permutation.txt	1234	Vishnu Shankar	NULL	IpWnGYDv7XvLEL71OOsbdw==
tuase	Staff	pseudo-random permutation.txt	1234	Vishnu Shankar	NULL	IpWnGYDv7XvLEL71OOsbdw==
Tuase	Doctor	pseudo-random permutation.txt	riversstate		Reject	anATwTmilwTBgilqgSWlrQ==
Tuase	Staff	pseudo-random permutation.txt	riversstate	7988	Reject	anATwTmilwTBgilqgSWlrQ==
vTest	Staff	Rsu.txt	12345	12345	NULL	AZDKvV7obci8RNR+OOD3kg==
vTest	Select Profession	Rsu.txt	1234	1234	Reject	gcGIKbMxgEstxp2HXXIZyQ==
vTest	Staff	Rsu.txt	today	1234	Reject	WA/ZOyU5/jTPfoiCh5JHrQ==
vTest	Staff	Rsu.txt	today	1234	Reject	WA/ZOyU5/jTPfoiCh5JHrQ==
vTest	Staff	Rsu.txt	12345	12345	NULL	AZDKvV7obci8RNR+OOD3kg==
vtest1	Staff	Rsu.txt	1234	1234	Reject	gcGIKbMxgEstxp2HXXIZyQ==
vtest1	staff	Rsu.txt	1234	1234	NULL	gcGIKbMxgEstxp2HXXIZyQ==
xtext	Doctor	Rsu.txt	4321	4321	NULL	Mwujuhf46RY1q7obk8hHHQ==

Figure 7: Encrypted File

Results and information in figure 7 are data that is store in big data cloud and the data in the field/column Value are the files encrypted data that are uploaded and encrypted. Similarly, domain authority can check and provide a permission to uploaded files haven the user's attributes and provide an authorization as it stated by data owner.

However, Table 2 lists the calculation effects of the method data file's numerical load. The computational load value obtained is dynamic, and varies with the efficiency of the CPU. Mean value is the average computational load, while the value of Stdev is the standard deviation of computational load, the Mean and Stdev values are thoroughly calculated in table 2. It shows the mean and standard deviation time of key distribution of secure data using our proposed solution with ECDH and AES

Table 2: The measuring result of file upload and download of big data cloud

Process	Algorithm	Data processed	Computational load		Notes
			Mean	Stdev	
Key distribution	*ECDH* (256-bit)	64 *bytes*	120.6 ms	48.6007 ms	Generating 32 bytes encryption keys
Secure the data	*AES* (256-bit)	64 *bytes*	307.9 ms	72.78652 ms	Encrypt 32 bytes messages.

Table 3: Calc. of Mean and Standard Deviation on computational load values

Experiment	Key Distribution (*ECDH*)	Secure the data (*AES*)
1	159 ms	273 ms
2	81 ms	315 ms
3	74 ms	313 ms
4	163 ms	174 ms
5	72 ms	350 ms
6	73 ms	340 ms
7	179 ms	176 ms
8	74 ms	327 ms
9	164 ms	331 ms
10	167 ms	180 ms
Mean	120.6 ms	307.9 ms
Stdev	48.6007 ms	72.78652 ms

Also, in table 3 we have the key distribution time and encryption time for all number of experiments thereby calculating the mean and standard deviation. The data in table 3 show the result of an experiment in ten iteration of key value generation and file upload time into big data cloud with mean value of 120.6ms of times in the experiments.

Performance Analysis of Propose Protocol: The HASBE and our protocol are implemented in Matlab. The simulation is implemented on a device running on Windows operating system, 4 GB RAM, and 2.30 GHZ dual-core CPU. From Figure 4.8 we can see that the key generation time cost is almost a constant value in our protocol. This is to say, the time cost of the main generation is kept almost unchanged with the rise in user numbers. Contrary to the key generation, the time cost of authentication is an exponential growth with the rise in the number of users. However, the time cost of the authentication is still within an acceptable level.

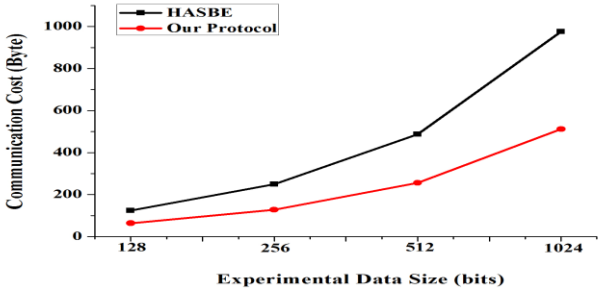

Figure 8: Communication cost comparison

Figure 8 shows that the communication overhead of the HASBE and our protocol increases with the growth of the amount of the experimental data. From figure 8, we can find that our protocol brings less communication cost to the system. Through the simulation results of the overhead, we can conclude that, compared to the HASBE, our protocol substantially improves the performance in terms of the computational and communication cost. The detailed description about the proposed big data authentication protocol and hierarchical attribute set based encryption with authorization structure mechanism is provided with the clear flow illustration. The principal object of this analysis is to; Provide a system-secure scheme so that only authorized users can log into the cloud to access Big Data storage files using a two-tiered hierarchical attribute authentication algorithm with a cloud-based Big Data Access Control System authorization structure that authenticates authorities or users.

i. Provide a data security scheme based on the use of the elliptic curve Diffie Hellman scheme that offers security features for forgery attack resistance, replay attack resistance and tree-based protected signature privacy protection and;
ii. The purpose of the hierarchical authorization attribute is to use two levels of Trusted Root Authority (TRA) and Domain Authority (DA). The TRA serves as an authorization manager and then is subordinated to the TRA by the DAs at second level.

Encryption time: This is known as amount of time taken by data owner to encrypt original data into encrypted data. Figure 12 indicates the encryption time of the new ECDH-AES for different data sizes. The encryption time is usually expressed in milliseconds and is measured as follows.:

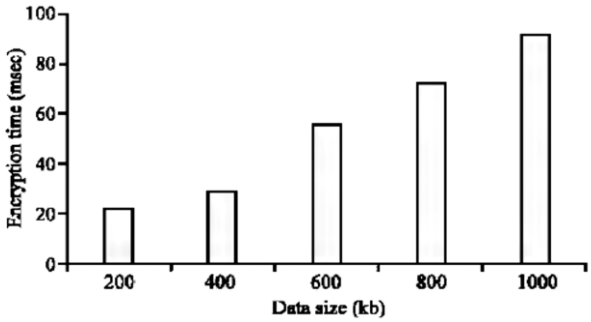

Figure 9: Encryption Time

Encryption time = Ending time - Starting time
From this illustration, it is analyzed that the encryption time can be increased with the linear increase in the data size.

Decryption time: Decryption time is defined as the amount of time taken by the data user to decrypt the encrypted data which is expressed in terms of milliseconds. It is calculated as follows:

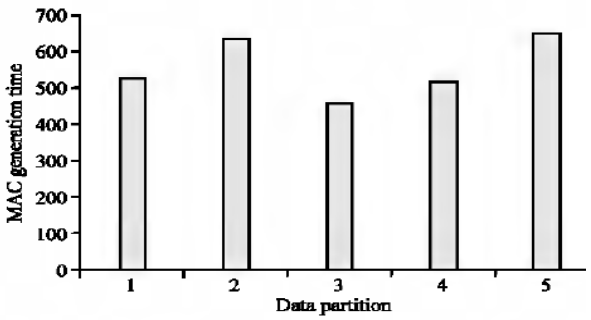

Figure 10: Memory Consumption

Decryption time = Ending time- Starting time

Memory consumption: memory use is defined as the volume of memory used for cloud data storage and is also known as the CPU's occupied capacity. Figure 10 shows the memory consumption of the proposed ECDH-AES technique with respect to the encrypted data partitions. It is calculated from this example that the proposed ECDH-AES allows the minimal memory space to store the data in the cloud server.

Comparative analysis: Table 4 contrasts the current scheme and alternative cryptographic methods for evaluating its performance depending on the execution times.

Table 4. Comparison between existing and proposed techniques

Algorithms	Number of user records	Execution time (sec)	Efficiency
KP-ABE	10	80	70
PROXY	10	120	60
Proposed Scheme	10	49	90

Security Proof of Propose Protocol: In our protocol, the hash function contains the identity information. If a person wants to pass authentication from his / her parent, he / she needs to randomly select a m for the collision resistant Hm. Hash is one-way tool, that is, it's not reversible. Hence, the tool can well secure the entity's identity when its parent verifies its authenticity. Our proposed authentication protocol (t,"/(n+1), m) is safe to resist forgery attack. Our protocol, with AES – ECDH, the authentication protocol will withstand assault against forgery. To prove the reliability, we evaluate it based on the collision resistant feature. The definitions of the collision-resistant method and the collision-finding algorithm that are set as follows:

Collision-resistant function: Suppose function F: -00{0, 1}k is the set of functions that are insensitive to (t, π)-collision-. The benefit of having a collision with any t-time algorithm can be referred to as Adv= Pr[F(1)=F(2)]<0. Here, it is generated randomly from π, i.e. {1} and 2, i.e. π.

Complexity Assumption: is the assumption that a certain computational problem within a cyclic group is difficult given a cyclic group G of order q thus the CDH assumption states that, given (g, g^a, g^b) for a randomly-chosen generator g and a, $b \in \{0\ldots, q-1\}$, it is computationally intractable to compute the value gab. The security of many cryptosystems is based on the this assumption, including notably the Diffie–Hellman key agreement scheme. The CDH assumption is related to discrete logarithm assumption, which holds that computing a discrete logarithm of a value base, a generator g is difficult. If taking discrete logs in G were easy, then the CDH assumption would be false: given (g, g^a, g^b) one could efficiently compute g^{ab} in the following way:

- compute a by taking the discrete log of g^b to base g;
- compute g^{ab} by exponentiation: $gab = (g^b)^a$;

Descriptions of the collusion-resistant function and the collision-finding algorithm are as follows:

Collusion-resistant function: Assume that the set of (t, π)-collusion-resistant functions is F:{0, 1}k equation. The value of any t-time algorithm that discovers a collision may be given as Adv = P r[F5-0(h)1) = Fπ(h)] <Here, it is generated randomly from π, i.e. {1} ubiquitous 2, i.e. π.

Table 6: Key generation Time vs No. of Attributes

No. of attributes	Existing (ms)	Proposed (ms)
0	0	0
5	0.18	0.1
10	0.3	0.19
15	0.43	0.32
20	0.625	0.38
25	0.752	0.452
30	0.875	0.603
35	1.1	0.712

V. CONCLUSION

We suggest a hierarchical attribute-based encryption with authorization structure framework authentication protocol which is inspired by the tree-based signature. The security analysis shows that our protocol can provide the resistance to forgery attacks and replay attack resistance, as well as the privacy preservation property. Our protocol can not only be used in the can two-level device situation but also well used in the hierarchical attribute authorization multi-level structure. We equate our protocol with the HASBE in the performance analysis.

A new security framework was introduced in this research work to allow secure data sharing which storing in a cloud environment. This system comprises four entities, such as data creator, CSP, TPA and user data. There, the digital owner uploads their application to a cloud before using the ECDH-ECC method to execute the key generation and encryption procedures. The main reason ECDH introduces is that the key witl1 produces improved complexity in an efficient manner, so the unauthorized user cannot easily access the key. In addition, the ECC is a highly secured encryption technique which encrypts the data based on the signature produced. The data owner then sends the encrypted data to the CSP, and generates the appropriate data signature. When the data recipient sends the file to the CSP, the CSP transfers the encrypted data to the user for access to the particular data. Correspondingly, the TPA shall forward the signature to the user of the data carrying out the process of authentication. If both the signatures created and received are the same the user must decrypt the data. The efficiency effects of current and proposed security mechanisms are tested in experimental assessment by using key generation time, encryption time, decryption time, computing overhead, and power utilization controls. Most solutions using attribute-based encryption (ABE) have been introduced in cloud computing for outsourced data access management; however, most of them are grappling with complex access control policies in implementation. While the huge profits generated by the cloud computing model are

promising for IT businesses, academic researchers, and prospective cloud customers, security issues in cloud computing turn out to be significant obstacles that will hamper widespread cloud computing implementations and activities in the future, without proper handling. One of the influential security concerns in cloud computing is data protection and privacy, thanks to its Internet-based data storage and management. Consumers must give up their data to the cloud service provider for storage and business operations in the cloud environment while the cloud service provider is generally a commercial enterprise that cannot be fully trusted. The results of this survey are as follows:

1. Man in the middle attack; the produced key is safe against the guy.
2. The protocol is based upon Diffie-Hellman problem being intractable.
3. It is capable of implicit key authentication, and key confirmation, it provides the recipient with assurance that he or she has calculated the valid key.
4. This is non-deterministic, using random numbers for session keys.
5. It's secure against the attacks discussed in this investigation.
6. Unless the assumptions are retained, then it can also be secured from attacks based on number theory.

Diffie-Hellman main algorithm exchange, is based on the premise that discrete logarithms are challenging to compute. The intractability theory is also the basis for a number of other public key systems being deemed safe. While substantial advancements have been made in discrete log algorithms over the last two decades, the discrete log still continues to be difficult in general. Sadly, there are no proofs of hardness in the area. As we discussed the proposed protocol is secure and its reliability is based on the problem of Diffie-Hellman and the complexity is the same. This technique is non-deterministic, as the encryption process requires randomization. For this work the protocol provided has several desirable security attributes. It is hoped that the protocol, or suitable modifications thereof, can be proven consistent for the security paradigm for big data under rational assumptions.

Recommendations: In the future our protocol will have lower overhead processing and connectivity, which will have a promising future in cloud big data authentication. We use advanced algorithms to encrypt and store data in cloud Big Data to protect the data. Therefore, the introduction of this methodology for the real-time device scenarios will improve this work in future. The neighborhood key generation process can also be done to improve safety standard.

REFERENCES

[1] Toninelli, A.; Montanari, R.; Kagal, L.; Lassila, O. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In Proceedings of the International Semantic Web Conference, Athens, GA, USA, 5–9 November 2006; pp. 473–486.

[2] Botta, A.; De Donato, W.; Persico, V. Pescapé, A. Integration of cloud computing and internet of things: A survey. Future Gener. Comput. Syst. 2016, 56, 684–700.

[3] Zissis, D.; Lekkas, D. Addressing cloud computing security issues. Future General. Computer. Systems. 2012, 28, 583–592.

[4] Bouabana-Tebibel, T.; Kaci, A. Parallel search over encrypted data under attribute-based encryption on the Cloud Computing. Comput. Secur. 2015, 54, 77–91.

[5] Akl, S.G.; Taylor, P.D. Cryptographic solution to a problem of access control in a hierarchy. ACM Transmission. Computer. Syst. 1983, 1, 239–248.

[6] Castiglione, A.; De Santis, A.; Masucci, B.; Palmieri, F.; Huang, X.; Castiglione, A. Supporting dynamic updates in storage clouds with the Akl–Taylor scheme. Inf. Sci. 2017, 387, 56–74.

[7] Akl, S.G.; Taylor, P.D. Cryptographic solution to a problem of access control in a hierarchy. ACM Transmission. Computer. Syst. 1983, 1, 239–248.

[8] Crampton, J.; Farley, N.; Gutin, G.; Jones, M.; Poettering, B. Cryptographic enforcement of information flow policies without public information via tree partitions 1. J. Computer. Security. 2017, 25, 511–535.

[9] Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.

[10] Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.

[11] Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Proceedings of the International Workshop on Public Key Cryptography, Taormina, Italy, 6–9 March 2011; pp. 53–70.

[12] Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.

[13] Lai, J.; Deng, R.H.; Li, Y. Expressive CP-ABE with partially hidden access structures. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, Seoul, Korea, 2–4 May 2012; pp. 18–19.

[14] Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Proceedings of the International Workshop on Public Key Cryptography, Taormina, Italy, 6–9 March 2011; pp. 53–70.

[15] Lee, C.-C.; Chung, P.-S.; Hwang, M.-S. A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments. IJ Netw. Secur. 2013, 15, 231–240.

[16] Li, Y.; Zhu, J.; Wang, X.; Chai, Y.; Shao, S. Optimized ciphertext-policy attribute-based encryption with efficient revocation. Int. J. Security. Its Appl. 2013, 7, 385–394.

[17] Hongbing, C., R Chunrning, H Kai, W. Weihong and L. Yanyan: Secure big data storage and sharing scheme for cloud tenants. China Communication., 12: 106-115, 2015.

[18] Sookhak, M., A Gani, M.K. Khan and R Buyya: Dynamic remote data auditing for securing big data storage in cloud computing. Inf Sci., 380:101-116.2017.

[19] Puthal, D., S. Nepal, R Ranjan and J. Chen: DPBSV-an efficient and secure scheme for big sensing data stream. Proceedings of the 2015 IEEE Conference on Trustcom/BigDataSE/ISPA Vol. 1, August 20-22, 2015, IEEE, Helsinki, Finland, ISBN:978-1-4673-7951-9, pp: 246-253, 2015.

