Journal of Autonomous Intelligence (2023) Volume 6 Issue 2
doi: 10.32629/jai.v6i2.566

ORIGINAL RESEARCH ARTICLE

A hybrid framework to enhance cloud security for storing and retrieving confidential data in clouds

N. Krishnaveni[1,*], C. Jayakumari[2]

[1] *Department of Computer Science, Bharathiar University, Coimbatore 641001, India*

[2] *Department of Computer Science, Middle East College, Muscat 113, Oman*

*** Corresponding author:** N. Krishnaveni, krishnavenivalliappan@gmail.com

ABSTRACT

The building pieces for creating a secure cloud framework for data exchange with authenticated and authorized users are ECC and ABAC. So, the goal of this study is to improve access control and encryption-related methods in order to increase security. The elliptic curve is a key component of the comparative study of cloud encryption and access control techniques. This research project's main goal is to provide a security architecture that combines authenticated access with attribute-based access control and better elliptic curve encryption. The second goal is to provide a better mapping strategy with reduced time and space complexity for elliptic curve encoding from plain text. To boost the performance of the standard ECC, a thorough algorithm focusing on designing an improved mapping mechanism for encoding plain text to elliptic curve points with excellent security has been included. The strength of the security should not be sacrificed in order to reduce security measures' overhead costs. ABE is regarded as an effective way for protecting cloud data, according to study results. Because to the use of complex pairing processes, the same is difficult to use. As a result, a hybrid approach using ECC and ABAC performs better to handle the increasing processing capacity.

Keywords: attribute based encryption; elliptic curve cryptography; attribute based access control; cloud security; security framework

ARTICLE INFO

Received: 24 April 2023
Accepted: 30 May 2023
Available online: 10 August 2023

1. Introduction

Cloud computing provides services over the internet on pay as per use basis. As the network connectivity and the dependency on online communication increased, large volumes of data is getting created. To save the expenses over storage devices, it is obvious that most of the data will be moved to cloud storage. Different types of cloud deployment models are public cloud, private cloud and hybrid cloud. Public cloud is operated and managed by providers whereas in private cloud the set up and management of cloud storage is to be done by the owner of the cloud. In hybrid cloud both the private and public cloud combination is used where data and applications are shared in between. To relieve from the difficulties of managing cloud set up, public clouds will be highly preferred compared to private cloud. Public cloud storage even on ordering storage, the responsibility for security is mostly vested with clients itself. Most of the cloud providers provide the security mechanism like encryption, authentication and access control policies, but its security is highly dependent on the vulnerability chances of provider and the tools over which the data is getting accessed too. The vulnerability lies inside the cloud provider, if the encryption and proper access control and authentication

mechanisms are built. Even though many of the clouds are being using many cryptographic schemes, many latest news on technology is still reporting that cloud is getting attacked. Hence a review has been done over the algorithms employed in existing cryptographic clouds. During the review, it is found that ensuring data security means to strictly maintain the confidentiality, integrity and availability of sored data in cloud with proper access control and authentication mechanisms. Many of the existing cloud security models keeps the confidentiality of the data but not able to ensure the authenticated and authorized access.

As the technology improves, the processing capability too becoming faster, may affect the security of encryption and access control mechanisms depended on mathematical hard problems. So the research on security measures will always be a field of research for enhancing the existing technologies. Encryption and authentication mechanisms are to be wisely chosen depending on the requirement of data management. Depending on the scenario where it is need to share the data to group of users those who have required attributes can opt for a common encryption key, even then it is extra secured using another secret so that to prevent sharing of keys with unauthorized users. Data privacy is another important factor for sensitive data and hence the access control policy is to be proper and the data in cloud must be encrypted, and also the decryption should be permitted only for the authenticated user. On observing the most commonly used public cloud storage Google drive, clients can control access to their data by setting access rights. The different options they provide on encryption are: one option is that the providers will be deciding encryption key and are stored with them and the end users need not bother about encryption or decryption; another option is customer can manage encryption key but keys are stored with provider and also the third option is customers can create and manage encryption keys and store the keys with them. Creating and managing keys is a tedious task for data owners and hence a support of a trusted agency can be used to create and manage the encryption keys will be quite useful and many cloud security models is making use of trusted agents and still the key management and secure key exchange is matter of research interest. Due to development of IoT the latency of the operations performed over the cloud data to be of greater concern and hence the need of decentralizing cloud storage is also highly in demand today.

Currently AES, ECC and ABE are widely used encryption techniques for sharing cloud data stored securely. The same secret key to be used for all users made the secret scheme not much efficient in distributing data among the whole community of users through non-secure channels. Hence, the ABE which helps users to regenerate secrets is more suitable for cloud data sharing. The most focused variants of ABE are KPABE and CPBAE. Among KPABE and CPABE, the mostly used ABE is CPABE. The main aim of sharing encrypted data with multiple authenticated users is possible with ABE but computational complexity made its practical implementation unfeasible. The computational cost is higher for attribute based encryptions as it uses high cost pairing operations. On considering the advantages of ABE, the review mainly covers the techniques adaptable with ABE which can reduce computational cost without correcting the required security directed to ECC. The reviews mention that ECC without pairing and ECC with pairing implement the ABE. ECC without pairing depended on the scalar multiplications to compensate the removal of pairing operations. Cloud service providers take advantage of massive computing resources spread across a wide geographic region to provide consumers with streamlined, powerful, and secure services at a low cost. The security concerns are shared by the CSP and the customer. Finally, certain elements of protection remain solely under the consumer's risk.

Customers may use a variety of Application Programming Interfaces (APIs) provided by cloud vendors to manage their cloud services. Unfortunately, not all APIs are completely stable. They may have been mistakenly considered to be, only to later be discovered to be vulnerable in some way. When the client organization adds its own application layer to the mix, the problem becomes even worse. Hence the research on applications securing cloud data is relevant in present world. The lack of trust in the cloud's security features is cited as one of the key obstacles and concerns that prevent users from entrusting their sensitive data to this

hazy entity called cloud. The two key issues that prevent a broader adoption and acceptance of cloud computing are information security and data protection.

2. Review of literature

2.1. Related works for ECC

Gbashi[1] suggested a map technique built on a genetic algorithm-generated casual matrix of letters and numbers, although it could only do encoding letters and digits. The identical letter won't be encoded differently in the system. Analyzing sample encoded and decoded values allows for the quick identification of similar patterns. Only when the mapped points are encrypted is security guaranteed.

Reyad[2] describes a map table in which the "ASCII" board is numbered in sequence to execute scalar multiplication on a selected basing point and the table is to be switched for doing the decode. Since there are only 128 total scalar multiplications, a brute force assault is feasible. The known plaintext attack is made feasible by repeated text being mapped to the same places.

The mapping approach used by Keerthi and Surendiran[3] transforms plain text typescripts into "ASCII" and then into hexa-decimal. Hexadecimal values are divided into 192 bit "x" with "y" groups. The benefit of this strategy is that no bit padding is necessary, but mapping will not protect against known plain text attacks.

For integer communications, Younes et al.[4] suggested a message map system paired with elgamal encrypting, and study also emphasizes the necessity of ECC for achieving excellent security standard. The map approach proposed by Almajed and Almogren[5] incorporates binary and decimal conversions as well as exclusive OR operations for blocks of plain text that may connect if the intrusive party has the initialization vector. Mahto and Yadav[6] compared the effectiveness of RSA and ECC and came to the conclusion that ECC was superior. The choice of ECC curves from an implementation viewpoint is discussed by Dhanda et al.[7] due to the main factors of small key size and network capacity. The authors' first comparison with the widely used symmetric key method AES revealed that ECC only behind AES in terms of execution speed while keeping all of the other essential security features. The study identified the requirement for an ECC solution that is efficient and secure against threats. The Weierstrass model, which determines elliptic curves in large prime fields, is emphasized throughout the paper's discussion of elliptic curves. For this procedure, you'll need a prime "P" equal to the size of the field, the elliptic curve's coefficients "a, b", a generator or base point for generating cyclic subgroups, "q" the order of "G", and a private key that is chosen at random from the range "1, 2, 3, ..., p − 1" while the public key is the private key G. The study discusses side channel attacks that are implementation-related, as well as hypothetical random walks and multiplicative group attacks on elliptic curves.

The implementation of scalar multiplications over elliptic curves and the various kinds of NIST approved elliptic curves are covered in Alimoradi et al.'s discussion[8]. ECC curves come in a variety of key lengths, including 192-bit (secp192r1-r stands for random), 256-bit (secp256k1-k and secp256r1-r stand for Koblitz), and many more. Weierstrass curves are quicker than Montgomery and Edward curves for performing elliptic curve operations such scalar multiplications, according to Gayoso Martinez et al.[9].

The requirement for employing ECC to create secrets based on groups as opposed to a single individual was suggested by Errahmani and Ikni[10]. The study examined Shamir's technique for sharing secrets, which requires a minimum amount of secrets to reconstruct the secret shared by various participants. In order to undertake verification before the required number of participants reconstitutes it using a polynomial matrix, elliptic curves with pairing are used. Users of the method may independently check information instead of working together. The suggested technique clarifies safe key exchange even if it takes more time.

Using brute-force on the key and a known plain text attack if the same text is mapped similarly, Naji et al.[11] studied ciphertext only attacks. In addition, Roy and Khatwani[12] evaluate the various cryptanalysis

techniques that may be used with ECC as well as reports of clogging assaults, pairing operations, and other in-depth operations. Using the use of elliptic curves and a linear secret sharing system, Ding et al.[13] suggested a pairing-free CPABE and evaluated whether overall efficiency had increased.

2.2. Related works for ABE

In their discussion of ABE's access control strategy for sharing data across dynamic groups, Belguith et al.[14] provided specifics. This study discusses privacy leaking and high decryption costs. The reviewers made note of the fact that most ABE access control policies are provided together with cipher text, which exposes the user's or data owner's personal information and compromises privacy. For the purpose of controlling key distribution, the key is stored in escrow with the designated third party. Due to bilinear pairing procedures performed by many authority servers, the large decryption cost was decreased by the approach. Another flaw in the proposed work is the administrative costs associated with controlling several servers. With KPABE, as opposed to CPABE, the key policy is linked to the private key, therefore the security of the cipher text primarily rests on the key issuer. As a result, the data owner has no control over who may read the cipher text. The individuals who have access to the cipher text may be approved by the data owners via CPABE.

A public cloud storage access model employing ABE was presented by Xue et al.[15] that allows cooperation amongst users with different attribute settings with the consent of the data owner. The authors suggested a compound policy tree to support this goal, but it is up to the reader to determine which undesirable cooperation should be prevented. The analysis shown above demonstrates how costly and time-consuming bilinear pairing calculations are when cloud data is shared across several users.

The difficulty of space is increased by cipher texts' bigger size compared to plaintext. Individual communications are encrypted using symmetric session keys in the technique presented by Sowjanya et al.[16] and each session key is encrypted using a pairing-free KPABE method employing ECC to minimize computational cost. Results showed that the certificate less public key encryption suggested by Lu and Li[17] provides computational benefits over pairing processes. The auditing technique suggested by Han et al.[18] is carried out without pairing calculations, and performance study confirms that the computational costs are decreased. By using scalar multiplications on an elliptic curve, Ding et al.[19] suggested a pairing-free CPABE based on OBDD. The number of valid pathways in the access policy determines the size of the cipher text. The experimental findings demonstrated the scheme's ability to withstand the selected plaintext attack and achieve computing efficiency. Hijawi et al.'s choice of attribute-based access policy and ECC for supplying security strength in the Internet of Things has their full endorsement (IoT)[20]. The authors drew attention to the key generation algorithm's vulnerability in such schemes.

The combination of ECC and CPABE without pairing operations was described by Sowjanya and Dasgupta[21], and the security is asserted using the Elliptic Curve Decisional Dille-Hellman assumption. Multi-factor authentication (MFA), which enables user authentication with the system both directly and over the cloud, was thoroughly surveyed by Ometov et al.[22]. MFA makes it possible for quick, simple, and reliable authentication when logging into a service, which is used for human-to-everything interactions. The most frequent characteristics utilized to match the person with his credentials are knowledge, ownership, and biometrics. MFA is recognized as the factor group that is often employed in high security applications and is based on biometric authentication. It is acknowledged as one of the crucial elements for verifying users' identities. The results of the survey show that, although being a more straightforward feasible means of identification, fingerprints should be backed by other authentication techniques. It is evident from the assessment that MFA has to be stored by a reputable institution. According to Anakath et al.[23], user-associated pseudorandom secrets are also regarded as one of the authentication factors.

2.3. Proposed framework

The proposed framework aims to create a secure cloud framework for data exchange by leveraging the combined strengths of ECC (elliptic curve cryptography) and ABAC (attribute-based access control). The primary objectives are to enhance access control and encryption methods to bolster security. The framework includes a security architecture that integrates authenticated access, attribute-based access control, and improved elliptic curve encryption. Additionally, a more efficient mapping strategy for encoding plaintext to elliptic curve points is developed to reduce time and space complexity. The framework emphasizes maintaining strong security measures without sacrificing performance. The study concludes that a hybrid approach combining ECC and ABAC is better suited to handle the increasing processing demands in cloud storage, as complex pairing processes pose challenges for ABAC implementation.

3. Novel maps scheme for ECC

3.1. Encoder for ECC

The conversion of plaintext characters to elliptic curve points using a unique technique is explained here. The suggested enhanced mapping approach makes advantage of security features like discrete logarithm problem-based scalar multiplications to guarantee that the same plaintext is encoded differently. The following improved encoding is suggested. Each character in plaintext is transformed into its value for "ASCII", and the point is appended to the value of "ASCII". After that, reverse the binary equivalent of the result to get the result's decimal equivalent. For the cubic equation of the elliptic curve, this decimal value will oblige as the coordinate "x". The value for the "y" coordinate is twice that of the "x" coordinate.

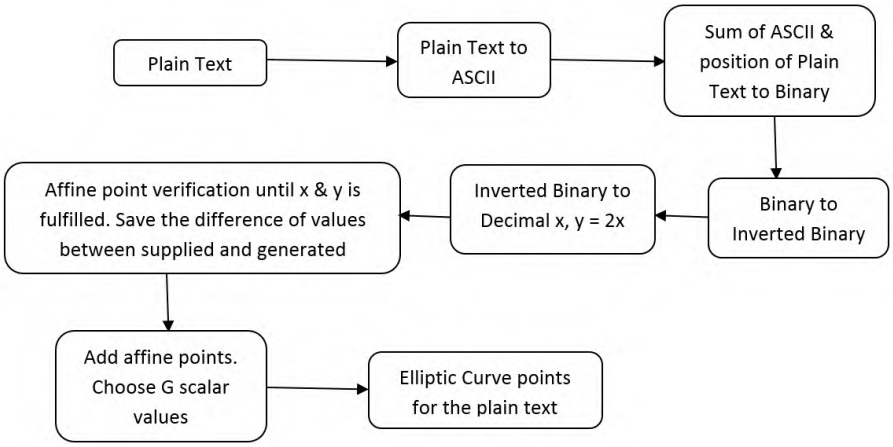

Figure 1. Encoding mechanism—plaintext to elliptic curve.

The algorithm for elliptic map will determine if the coordinates for input and satisfy the elliptic curve. If not possible, set a high number for the counter, increase "x" and "y", check the value for elliptic curve cubical, and continue the process until the equation's coordinates are created. The difference between the original "x" value provided and the final "x" value received is recorded separately. To improve security by preventing it from being restricted to ASCII values, the base point G's selected scalar value must be appended to the newly formed elliptic curve point. The methods for translating input to encoded text are shown in **Figure 1**. The suggested mapping approach works well for applications that want more protection. The suggested approach assures security by generating unique cypher text even when all text blocks in a single document have the same scalar values. Different scalar values for different blocks will increase the cost of communicating with consumers. Intruders may find it more challenging to exploit the mapping technique if big "p" values are used for the elliptic curve.

Encoder algorithm for ECC

Input: Plaintext
Output: Elliptic curve points
encoding ():
Generate global parameters p, a, b, G of selected elliptic curve
Initializing x -> 0, y -> 0 and n integers for scalar Gmatrix
while (x<n)
x->0, y->0
equivalentascii [x]->message[x]_plain text to ASCII
binarray [x] -> tobinary(equivalentascii[x] + x)_binary equivalent of sum of
position and ASCII
invarray[x] -> inverse (binarray[x])_convert 0 to 1 and 1 to 0
decimalarr [x] -> todecimal (invarray)_binary to decimal conversion
xymatrix[x][y] -> decimalarr[x],2*decimalarray[x]_dummy x, y coordinates
epmatrix [x][y] -> ellipticmapper(xymatrix[x][y])_elliptic curve points
di
erence [x] -> xymatrix[x] - epmatrix [x] _difference of supplied and actual coordinates
g_matrix[x][y] -> (s[x])(Gx,Gy)_scalar 'G matrix
mapped_list [x][y] -> Add(epmatrix, g_matrix)_point addition
x -> x + 1, y -> y+1
end while
Return mapped_list, g_matrix, difference
End FUNCTION

3.2. Decoder for ECC

Figure 2's depiction of decoding shows how it works in reverse of encoding. The interchange of scalar and difference values is necessary for decoding. The encoded values and scalar "G" values are subtracted, and the resulting matrix is added to the trapdoor difference values. The process is then converted to binary, inverted binary, and lastly the decimal equivalent. To create plain text, subtract the location and map the remaining data using an ascii table. In comparison to other current ways of mapping schemes, this system is practical for secure document encoding with minimal overhead during transmission. On randomly selected values for multiplication with the generator point used in the addition operation of the mapping scheme, the security of the mapping scheme relies.

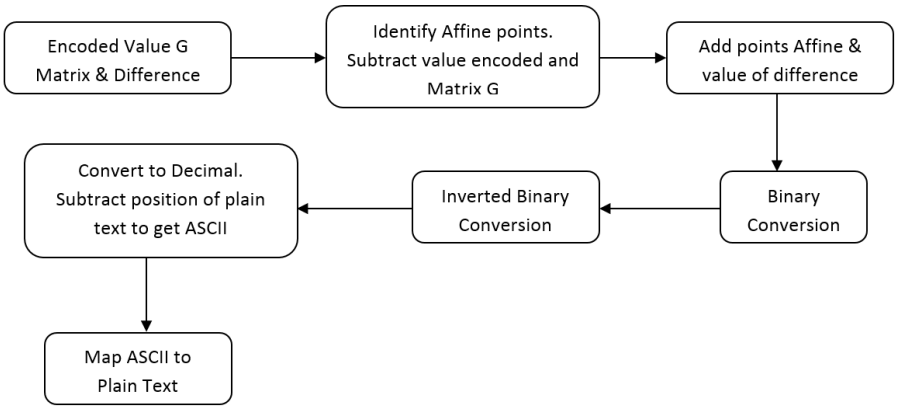

Figure 2. Decoding mechanism—elliptic curve to plaintext.

Decoder algorithm for ECC

Input: mapped_list, g_matrix, difference
Output: Plaintext
FUNCTION Decode ():
Initializing x -> 0, y -> 0
while (x < n)
x -> 0, y -> 0
epmatrix[x][y] -> mapped_list [x][y]- g_matrix[x][y]
decarray [x] -> epmatrix[x]+ di
erence[x]
binarray[x] -> binary (decarray[x])
binarray[x] -> inverse(binarray[x])
decimal[x] -> todecimal(binarray[x])
asciiarray [x] -> toascii(decimal[x])
x -> x +1, y -> y+1
end while
Return mapped_list, g_matrix, difference
end FUNCTION

4. Attribute based access control scheme

Data sharing via the internet must be done while maintaining data security and privacy. User authentication and allowed access are crucial for maintaining the security and privacy of cloud data. IAM (identity and access management) uses access control and identity management rules to govern user access. IAM provides data security by limiting who has access to what information. IAM is to be used for both authentication and authorization. More security controls are provided by IAM when it is implemented properly. Usernames and passwords are two identity management tools that are often used. IAM began using multifactor authentication, such as biometrics, due to the rise in security concerns, including the theft of passwords. In order to avoid impersonation, authentication criteria relating to the user will be compared to the identity database.

Identifying the change in access privileges and processing in accordance with it is the bigger issue in IAM. IAM is crucial in cloud computing since internet access is required to access distant cloud data. Enforcing access control is a part of access management. Access control allows authorized users to access resources and forbids unauthorized ones from doing so. Security systems should be managed using secure identification and access control. Access management enables authorization, which is the focus of identity management. A user's access to the data is verified by authorization. Software for identification and access control on servers and mobile devices should be compatible.

Depending on how sensitive the shared data is, the organization will choose the kind of access control. With discretionary access control, the access control policy is determined by the data owner. Depending on the users' roles, role based access control (RBAC) limits access. ABAC makes advantage of the ambient qualities as well as the properties of objects or subjects, which are resources or users, respectively. Also, cloud services use it quite well. Given the values of the characteristics of the subject, object, or environmental circumstances, a policy is a representation of the rules that decide whether a requested access should be permitted. In identity and access management, ABA and RBAC may coexist. With the addition of characteristics, rules, and context, RBAC has enhanced the authorization process. ABAC facilitates control organization based on data properties. Although ABAC employs a mix of user characteristics, resource

attributes, and environmental attributes, RBAC is primarily based on roles. RBAC limits access to system activities, while ABAC might limit access to data. As a result, ABAC offers finer grained access than RBAC. The term "ABAC policy" refers to access control policies that are based on characteristics, which might be based on persons, objects, or the environment. ABAC is described as "an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions", by NIST. The ABAC guarantees that users only have access that is permitted and guards against both attackers and authorized users abusing their rights. The rules created for ABAC signify more comprehensive user information. ABAC rules are assessed in real-time depending on the supplied characteristics. Here, the ABAC policy decides whether the subject or user requests access depending on the provided characteristics. The fact that the ABAC rules may be developed and maintained without direct input from the prospective users is by far the biggest benefit. RBAC may be migrated to ABAC in order to provide more granular access, and ABAC can support RBAC. Formal agreements between attribute suppliers and customers should adhere to the necessary information protection requirements for safe communication. The system performs better when there are fewer characteristics. To guarantee the accuracy of attribute values pertaining to the registered users, attribute administration should be carried out by an attribute authority. ABE encodes cipher text with data access control rules and controls. The cryptographic enforcement of access control regulations utilizing pairing operations, which is quite expensive, is one of the main uses of ABE. Depending on the ABE version being utilized, key policy attribute based encryption (KPABE) and CPABE correlate attributes and access structures with keys and cipher text. **Figure 3** shows the ABAC policy's organizational structure.

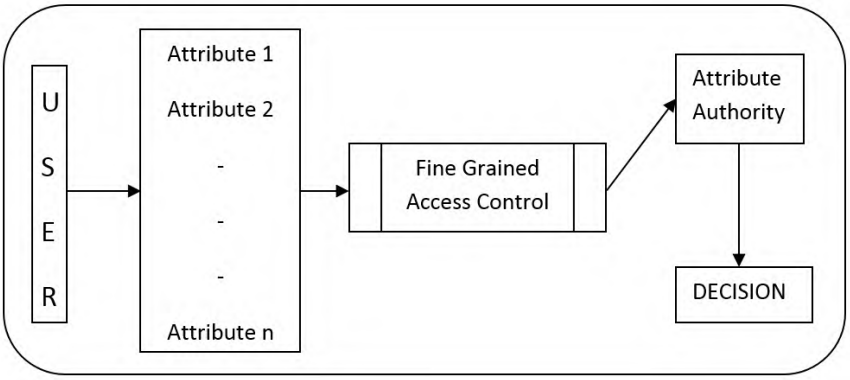

Figure 3. Attribute based access control mechanism.

Implementing ABAC requires attribute management and enrolment. One way to implement ABAC is with Boolean logic. A trusted server known as the attribute authority oversees attribute-based access in the cloud security concept. In order to ensure user authentication, user-id, and fingerprint registration is included, data users must register with the attribute authority by submitting their attributes. The attribute authority that oversees the data owner's policy then navigates the policy and makes a decision. The authorization of access to an authenticated user is indicated by the granting of permission. As the fingerprint serves as a powerful authentication element, impersonation attempts by attackers are challenging. Access structure may be implemented in a variety of methods, including via threshold gates, AND gates, and ordered binary decision diagrams (OBDD). Either AND gates and OR gates combined at the interior node or threshold gates themselves as interior nodes are employed for access structure implementation of threshold gates. Attributes are found in its leaf nodes.

Hence, only monotone access structures are allowed for this access structure. Threshold values are used to explain Boolean functions that are represented by weights over a monotone access structure. The fact that

non-monotone calculations are utilized in monotone functions is indicated by the addition of its positive weights and checking whether it is up to a threshold. As a result, it was discovered to be useful in cryptography for secret sharing methods. Weight serves as a proxy for each user's share size in Shamir's top-secret sharing system. The strategy becomes very inefficient if the size of the shares is exponential in the quantity of users, *n*. (*t*, *n*), where $t = n$ for AND gates, is a symbol for threshold gates. The approach used shares a secret component with the policy's properties, and Lagrange interpolation enables its reconstruction. Rows are used to represent characteristics in the linear secret sharing scheme (LSSS), and column vector matrices are multiplied to share secrets among the rows. When ABE is combined with secret distribution and creation utilizing LSSS and threshold gates, the exponential operation and hence its computing cost are increased.

5. Implementation of the proposed ECC method

The Pivotal cloud foundation (PCF) based private cloud of Intel Core i7-5005U CPU @ 4.00 GHz processor with 16 GB RAM and 64 bit Linux OS, X-64 based processor is used to implement the suggested ECC technique. Using an elliptic curve as the basis for the experiment, the scalar values selected for the two blocks of five characters each are 1, 2, 3, and 5, with the parameters "a as 1", "b as 1", "p as 23", and "G as (0, 1)". The encoded points for the example plaintext "world" using the keys 1, 2, 3, and 4. Plain text "w" has been encoded as (18, 20), "o", "12, 19", "r", "12, 4", "l", and "d" (7, 11). It is clear from the encoded points that various plaintext characters are randomly encoded to elliptic curve points of the selected curve. By picking a high bit curve for big plaintext, the distribution of the plaintext to different elliptic curve points is randomized.

The time required to multiply an elliptic curve is under one second. For encoding and decoding the I letter in plain text, the computational cost is computed based on the quantity of scalar multiplications, number of point additions, and number of point subtraction of elliptic curve points. Let "M" stand for the cost of scalar multiplication, "ADD" for the price of adding elliptic curve points, and "SUB" for the price of subtracting two numbers. The table below lists how many addition and scalar operations will be performed using the suggested encoding and decoding strategy over I plain text characters.

5.1. Evaluation by scalar multiplication

The below Equations (1)–(6) give the evaluation outcome of the scalar multiplication using the proposed ECC method.

$$\text{Encoding Computation Cost} = M\ (5) \tag{1}$$

$$\text{Decoding Computation Cost} = M\ (5) \tag{2}$$

$$\text{Encryption Computation Cost} = M\ (i + 1) \tag{3}$$

$$\text{Decryption Computation Cost} = M\ (1) \tag{4}$$

$$\text{Total Computation Cost for encoding and encryption} = M\ (5) + M\ (i + 1) \tag{5}$$

$$\text{Total Computation Cost for decoding and decryption} = M\ (5) + M\ (1) \tag{6}$$

5.2. Evaluation by addition or subtraction

The below Equations (7)–(12) give the evaluation outcome of the scalar multiplication using the proposed ECC method.

$$\text{Encoding Computation Cost} = ADD\ (i) \tag{7}$$

$$\text{Decoding Computation Cost} = SUB\ (i) \tag{8}$$

$$\text{Encryption Computation Cost} = A\ (i) \tag{9}$$

$$\text{Decryption Computation Cost} = SUB\ (i) \tag{10}$$

$$\text{Total Computation Cost for encoding and encryption} = 2\ (A\ (i)) \tag{11}$$

$$\text{Total Computation Cost for decoding and decryption} = 2\ (SUB\ (i)) \tag{12}$$

5.3. Outcome of proposed ECC

The computational cost expressed in terms of scalar and addition operations on the elliptic curve and illustrated in the equations shows that the proposed method's decryption cost is lower than its encryption cost. The system requires 127 scalar multiplications before encoding, however the bulk of these operations may be rendered pointless if the size of the raw text is tiny[2]. In the suggested system, scalar multiplications rely on the size of the plaintext; as a result, computation costs are lower for smaller inputs. A pair of elliptic curve coordinates are produced after applying ECC on a plain text string corresponding to a single plain text letter. Instead, when encrypting using the suggested mapping approach, only "n + 1" elliptic curve coordinates are produced for "n" plaintext letters. As a result, the space complexity of the cryptographic system is reduced by this approach's reduction in the size of the ciphertext. In the experiment, P192 elliptic curves are used.

6. Evaluation of the hybrid method (ECC + ABAC)

The experiment employs the P192 elliptic curve and the domain settings listed below.

gx, gy = (602046282375688656758213480587526111916698976636884684818,
174050332293622031404857552280219410364023488927386650641)
a = −3
b = 2455155546008943817740293915197451784769108058161191238065
p = 6277101735386680763835789423207666416083908700390324961279
q = 6277101735386680763835789423176059013767194773182842284

Figure 4 illustrates the estimated encryption and decryption times of example plain text sizes, including encoding and decoding times, using the P192 elliptic curve.

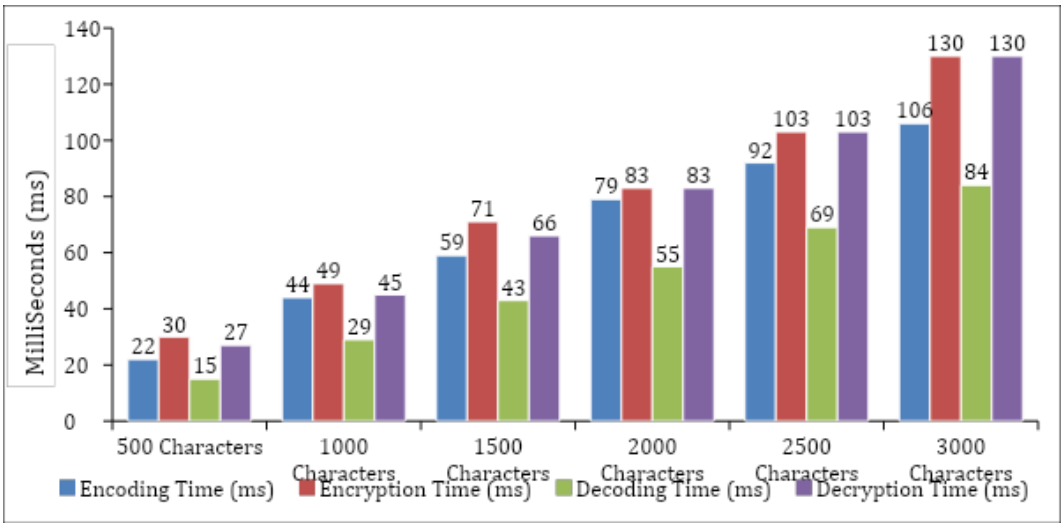

Figure 4. Evaluation of encoding, encryption, decoding and decryption time in milliseconds (ms).

The experimental data demonstrates that encryption and decryption time rely on the amount of plain text and that such operations require relatively little time compared to the level of protection they provide. **Figure 5** and **Table 1** show the size of the cipher text, which is equal to the plain text size of 409 characters, as well as the encryption and decryption times used by the proposed system and alternative ECC NIST certified P192 curve systems. In comparison to existing elliptic curve cryptographic techniques that combine a specific encoding and decoding methodology with encryption and decryption, the suggested system creates a relatively little quantity of cipher text. Using 409 plaintext characters as input, the encryption and decryption times of several techniques were compared to the proposed scheme. According to the experimental findings, the

suggested technique requires less time for encryption and decryption than previous schemes, as well as less storage space for cipher text.

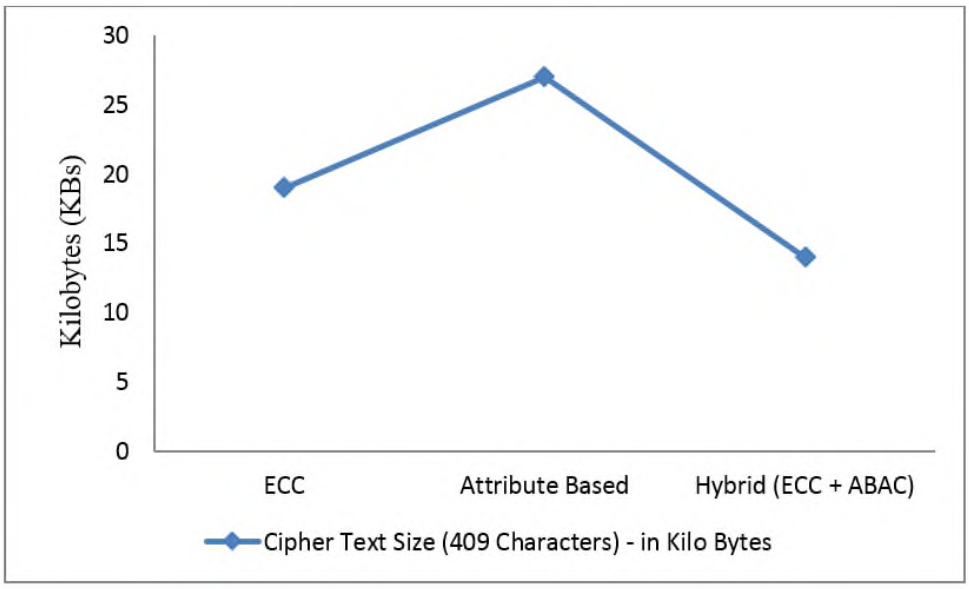

Figure 5. Evaluation of cipher text size in kilobytes (KBs) for the proposed method with ECC and attribute based methods.

Table 1. Cipher text size in kilobytes (KBs) for 409 characters.

Method	Cipher text size (409 characters)—in kilo bytes
ECC	19
Attribute based	27
Hybrid (ECC + ABAC)	14

ECC and ABAC are two security concepts that can be used together to improve security. ECC is a type of cryptography that offers robust encryption and key exchange methods. It ensures secure communication channels and protects data during transmission. On the other hand, ABAC is an access control model that enables precise access control decisions based on attributes associated with users, resources, and environmental conditions. It allows for flexible and granular authorization policies by considering various attributes. The relationship between ECC and ABAC lies in their complementary nature. ECC provides the encryption and key exchange mechanisms needed to secure communication channels and protect data integrity. It prevents unauthorized access and eavesdropping. ABAC, on the other hand, enables fine-grained access control by considering attributes such as user roles, resource classifications, and environmental conditions. This allows organizations to define dynamic and context-aware access control policies. By combining ECC and ABAC, organizations can achieve a higher level of security. ECC ensures secure communication, while ABAC provides the means to define and enforce access control policies based on attributes. This combined approach allows for stronger protection of sensitive information, prevents unauthorized access, and ensures compliance with security requirements in various domains (**Table 2**).

Table 2. Evaluation of security efficiency.

Method	% of security efficiency
ECC	91%
Attribute based	92.8%
Hybrid (ECC + ABAC)	96.3%

The ECC method achieves a security efficiency of 91%. ECC demonstrates a high level of effectiveness in securing data and communication channels. The ABAC method achieves a security efficiency of 92.8%. With a security efficiency of 92.8%, ABAC demonstrates a high level of effectiveness in controlling access to resources based on attributes. The hybrid approach, which combines ECC and ABAC, achieves a security efficiency of 96.3%. This approach leverages the strengths of both ECC and ABAC to enhance security. By integrating ECC's encryption and key exchange mechanisms with ABAC's attribute-based access control, the hybrid approach demonstrates a higher security efficiency of 96.3%. This suggests that the combined use of ECC and ABAC provides a more robust and effective security solution.

7. Conclusion

The research's goals of improving the cloud storage paradigm for safely storing and sharing data across active groups online have been met. High end security, especially for brief communications, is provided via novel access control and ECC algorithms without incurring significant computing costs. While sending communications, this kind of security improvement is advantageous since it considers user qualities rather than identification when encrypting the data. The security was strengthened by having many secrets linked to decrypting cipher text. The developed model's security enhancements included the less-expensive encoding of plain text to elliptic curve points. The same plain text being encoded to distinct elliptic curve points at various places demonstrates the great level of security provided. The suggested technique decreased the quantity of cipher text and the time complexity of the encryption and decryption processes, according to a comparison of experimental tests conducted using various mapping systems. The findings of the proposed mapping scheme's crypto-analytical analysis guarantee its security against various security intrusions. The data owner is supported by the trusted server for maintaining characteristics, which also lessens the burden involved in calculating the secret key and public key for encryption. The remainders of the cipher texts, which need less storage, are preserved with attribute authority and just a portion is stored in the cloud. By employing the secrets of verified users to execute holomorphic encryption on only one part of cipher texts, attribute authority improved key exchange security and never generated a significant computational burden. The characteristics of registered data users are used to navigate the access control policy, and when they reach the leaf node with the value "1", holomorphic encryption is carried out. The exchange of secrets is followed by the computation of the real secret key by the data user, followed by the decryption operation. The key that is sent over the network to data consumers is not the real secret key; instead, users calculate the real secret key from it to decode the cipher text. The created model may be used to share data via edge nodes in a decentralized computing environment.

Author contributions

Conceptualization, CJ; methodology, NK; software, NK; validation, NK; formal analysis, CJ; investigation, CJ; resources, CJ; data curation, CJ; writing—original draft preparation, NK; writing—reviewing and editing, CJ; visualization, NK; supervision, CJ; project administration, CJ.

Conflict of interest

The authors declare no conflict of interest.

References

1. Gbashi EK. Proposed secret encoding method based genetic algorithm for elliptic curve cryptography method. *Iraqi Journal of Information Technology* 2018; 8(3): 21–46.
2. Reyad O. Text message encoding based on elliptic curve cryptography and a mapping methodology. *Information Sciences Letters* 2018; 7(1): 7–11.

3. Keerthi K, Surendiran B. Elliptic curve cryptography for secured text encryption. In: Proceedings of 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT); 20–21 April 2017; Kollam, India. pp. 1–5.
4. Younes L, Youssef A, Saiida L. Definition and implementation of an elliptic curve cryptosystem using a new message mapping scheme. In: Proceedings of the 3rd International Conference on Networking, Information Systems & Security; 23–25 March 2020; Athens, Greece. pp. 1–6.
5. Almajed HN, Almogren AS. SE-Enc: A secure and efficient encoding scheme using elliptic curve cryptography. *IEEE Access* 2019; 7: 175865–175878. doi: 10.1109/ACCESS.2019.2957943
6. Mahto D, Yadav DK. Performance analysis of RSA and elliptic curve cryptography. *International Journal of Network Security* 2018; 20(4): 625–635. doi: 10.6633/IJNS.201807 20(4).04)
7. Dhanda SS, Singh B, Jindal P. Demystifying elliptic curve cryptography: Curve selection, implementation and countermeasures to attacks. *Journal of Interdisciplinary Mathematics* 2020; 23(2): 463–470. doi: 10.1080/09720502.2020.1731959
8. Alimoradi R, Arkian HR, Razavian SMJ, Ramzi A. Scalar multiplication in elliptic curve libraries. *Journal of Discrete Mathematical Sciences and Cryptography* 2021; 24(3): 657–666. doi: 10.1080/09720529.2017.1378411
9. Gayoso Martinez V, Hernández Encinas L, Martín Muñoz A, Durán Díaz R. Secure elliptic curves and their performance. *Logic Journal of the IGPL* 2019; 27(2): 277–238. doi: 10.1093/jigpal/jzy035
10. Errahmani HB, Ikni H. Verifiable self-selecting secret sharing based on elliptic curves. *International Journal of Software Innovation (IJSI)* 2020; 8(3): 51–68. doi: 10.4018/IJSI.2020070104
11. Naji MA, Hammood DA, Atee HA, et al. Cryptanalysis cipher text using new modeling: Text encryption using elliptic curve cryptography. *AIP Conference Proceedings* 2020; 2203(1): 020003. doi: 10.1063/1.5142095
12. Roy S, Khatwani C. Cryptanalysis and improvement of ECC based authentication and key exchanging protocols. *Cryptography* 2017; 1(1): 9. doi: 10.3390/cryptography1010009
13. Ding S, Li C, Li H. A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT. *IEEE Access* 2018; 6: 27336–27345. doi: 10.1109/ACCESS.2018.2836350
14. Belguith S, Kaaniche N, Laurent M, et al. PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT. *Computer Networks* 2018; 133: 141–156. doi: 10.1016/j.comnet.2018.01.036
15. Xue Y, Xue K, Gai N, et al. An attribute-based controlled collaborative access control scheme for public cloud storage. *IEEE Transactions on Information Forensics and Security* 2019; 14(11): 2927–2942. doi: 10.1109/TIFS.2019.2911166
16. Sowjanya K, Dasgupta M, Ray S, Obaidat MS. An efficient elliptic curve cryptography-based without pairing KPABE for Internet of Things. *IEEE Systems Journal* 2019; 14(2): 2154–2163. doi: 10.1109/JSYST.2019.2944240
17. Lu Y, Li J. Constructing pairing-free certificateless public key encryption with keyword search. *Frontiers of Information Technology & Electronic Engineering* 2019; 20(8): 1049–1060. doi: 10.1631/FITEE.1700534
18. Han J, Li Y, Chen W. A lightweight and privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities. *Computer Standards & Interfaces* 2019; 62: 84–97. doi: 10.1016/j.csi.2018.08.004
19. Ding S, Cao J, Li H. Efficient pairing-free CP-ABE based on ordered binary decision diagram. *Journal on Communications* 2019; 40(12): 1–8. doi: 10.11959/j.issn.1000-436x.2019234
20. Hijawi U, Unal D, Hamila R, et al. Performance evaluation of no-pairing ECC-based KPABE on IoT platforms. In: Proceedings of 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT); 2–5 February 2020; Doha, Qatar. pp. 225–230.
21. Sowjanya K, Dasgupta M. A ciphertext-policy Attribute based encryption scheme for wireless body area networks based on ECC. *Journal of Information Security and Applications* 2020; 54: 102559. doi: 10.1016/j.jisa.2020.102559
22. Ometov A, Bezzateev S, Makitalo N, et al. Multi-factor authentication: A survey. *Cryptography* 2018; 2(1): 1. doi: 10.3390/cryptography2010001
23. Anakath AS, Rajakumar S, Ambika S. Privacy preserving multi factor authentication using trust management. *Cluster Computing* 2019; 22(5): 10817–10823. doi: 10.1007/s10586-017-1181-0