Multiauthority KP-ABE Access Model with Elliptic Curve Cryptography

A. Ferrer-Rojas , B.T.J. Maharaj

***Abstract*—The rapid and expansive integration of Internet of Things (IoT) environments across various industrial sectors has led to an unprecedented surge in data generation and management. This exponential growth in data underscores the critical necessity for robust data security methodologies that can effectively safeguard the confidentiality and integrity of information without imposing undue computational burdens. In response to this challenge, numerous studies have sought to leverage Attribute-Based Encryption (ABE) as a means to enable fine-grained access control. Among the ABE variants, Ciphertext Policy ABE (CP-ABE) and bilinear pairings have emerged as popular choices to construct security schemes that strike a balance between robust protection and computational efficiency.**

Despite the advancements achieved through CP-ABE and bilinear pairings, a prevalent concern arises in the utilization of Linear Secret Sharing Scheme (LSSS) access policies. LSSS policies, while providing a flexible and expressive way to define access controls, can significantly impact the execution time of encryption methods. This study recognizes the importance of addressing this challenge and explores the potential of employing a Key Policy Attribute-Based Encryption (KP-ABE) approach. The primary objective is to mitigate the computational overhead associated with encryption methods, thereby enhancing the efficiency of data security measures within IoT environments.

Furthermore, this research delves into the incorporation of Elliptic Curve Cryptography (ECC) to generate cryptographic keys. ECC, known for its strong security properties and computational efficiency, is considered a promising approach to bolster data security while concurrently minimizing computational overhead. By integrating KP-ABE with ECC, this study aims to offer a comprehensive solution that ensures robust security measures within the intricate landscape of IoT environments.

Through detailed analysis and empirical investigation, the research endeavors to contribute valuable insights to the ongoing discourse on securing IoT data in a manner that aligns with the dual imperatives of security and computational efficiency.

***Index Terms*—ABE, ECC, IoT, LSSS Access Policiy, Multiauthority**

I. Introduction

INERNET of Things (IoT) environments have gained a great deal of traction in recent years [1]. Sectors such as healthcare [2], agriculture [3] and power generation [4] have adopted IoT environments to automate and improve their operations. However, one of the consequences of this is that IoT environments are also becoming larger, containing more devices which produce more data [5], [6]. This increase in data has increased the computational overhead of traditional data security methods used in IoT environments. Emphasis has been placed on developing low computationally intense methods of ensuring data security. This includes attempting to decrease total execution time of data security procedures and reducing the text size of messages sent. There is also an emphasis on creating more robust data security methods, as oppositions in data security are only improving.

All data security schemes in IoT systems must incorporate a public key infrastructure (PKI) [7], as traditional symmetric key infrastructures are too unreliable in systems with as many entities as IoT systems. There have been attempts at creating hybrid structures which utilize symmetric and asymmetric methods, such as digital signature schemes. One example of such a scheme is the work produced by C. Meshram et al. [8]. This scheme utilized one way hash functions and multi-variable signatures with extended chaos maps to provide provable security. This scheme outperformed other signature schemes in both signing execution time and verification time. These improvements are significant with a performance benefit of up to 8500ms in total execution time.

However, the literature landscape has preferred the use of pure PKI, such as the work conducted by P. Liu [9]. P. Liu had researched the use of indistinguishability obfuscation and weak related key attack (RKA) secure one-way functions to ensure a scheme that is Related Randomness Attack (RRA) model secure. Two constructions were produced and analysed, however, empirical analysis of the computation time of both constructions were not considered in the study. Only the theoretical analysis of both constructions was considered.

Some other popular methods for creating more lightweight data security schemes choose to utilize access structures based on Role Based Access Control (RBAC) [10], or Fine Grain Access Control [11]. RBAC structures assign roles to users in the system, and based on these roles the user has access to certain data. This allows for flexible access to data but typically does not provide as robust security as other methods. Fine Grain Access Control utilizes attributes instead of roles and is much more restrictive and secure than RBAC structures, but do not allow for privacy preserving access or enhanced scalability. There are other popular methods such as Mandatory Access Control (MAC) and Discretionary Access Control (DAC), but none of these methods are prevalent as Attribute Based Encryption (ABE) is in current literature.

ABE is the most flexible of the access structures provided. It allows designers to choose between symmetric, asymmetric and homomorphic cryptography methods. It provides robust

levels of security, but it is hindered by its complexity. Things such as complex key management, performance overhead, security assumptions in the proof of security, and complexity of implementation must be considered when designing of ABE based security models. With the addition of new methods such as edge-computing and lightweight cryptography, the possibilities are endless.

There have been attempts at combining ABE with searchable encryption, such as the work produced by I. Huso et al. [12]. This research focused on addressing the challenges in effective and privacy-preserving data dissemination within the context of Multi-Access Edge computing for the Industrial Internet of Things (IIoT). The combination of searchable encryption and ABE allowed for researchers to ensure data confidentiality, flexible protection against unauthorized access, and privacy-preserving data dissemination directly at the network edge. This system looked to incorporate Multi-Access Edge computing applications, specifically through the use of Trapdoors sent to Edge Servers. This resulted in a system which did not show significant bottleneck as the size of the system was increased, and created a secure yet efficient data security scheme. Zhiguo Wan et al. [13] decided to make use of the hierarchical structure of IoT environments which utilize edge-computing to modify their access structure. This resulted in an ABE data security model which had a high level of security, flexibility and did not compromise efficiency to do so. This was made possible with their use of a recursive set based key structure and an AND gate access structure.

Xiong Li et al. [14] attempted using decryption outsourcing, another edge-computing method, in an attempt to create a more efficient data security scheme. The group also managed to address the issues of key escrow and attribute revocation in their model. Having the outsourced decryption being performed by the edge-server also allowed for resistance to collusion attacks. When compared to other schemes, the data security scheme present here performed very well in encryption and decryption time.
Similarly, J. Li et al. [15] also attempted outsourcing the decryption of devices in the IoT environment. Their method was to delegate all access policy and attribute-related operations during key generation or decryption to a Key Generation Service Provider (KGSP) and a Decryption Service Provider (DSP), respectively. This results in only a constant number of straightforward operations to be executed by the attribute authority and eligible users. Their outsourced decryption was tested against that of other outsourcing decryption schemes and found to be as efficient as the selected schemes while providing more robust security. The research also found that the key generation execution time of the system was decreased, although this is a rarely applicable benefit.

There are also other methods of decreasing computational complexity in ABE schemes, such as the research done by Vanga Odelu and Ashok Kumar Das [16]. This group created a constant key size ABE scheme, which utilized Elliptic Curve Cryptography (ECC). ECC is a popular lightweight cryptography methods, which provides higher levels of security at lower key sizes. While this scheme did not perform well in decryption time compared to its contemporaries, it did outperform them in encryption time and total time of execution. It did so while still having a smaller key size than all other schemes it was compared to.

The research done by Sangjukta Das and Suyel Namasudra [17]. Utilized CP-ABE with ECC and Linear Secret Sharing Scheme (LSSS) to produce a fine-grain access control model with reduced computational overhead. The system managed to outperform many contemporaries in total execution time while providing robust security against multiple malicious attacks. However, this scheme performed poorly in terms of execution time when performing encryption methods. This is most likely due to the design decision of using a CP-ABE, where the Data Owner is responsible for establishing the access policy used to determine which Data Users may have access to the data.

This study looks to improve upon the work done by Sangjukta Das and Suyel Namasudra [17], by converting the scheme to a Key Policy based ABE (KP-ABE). The main drawbacks of this conversion would be the increased computational overhead of the Key Generation and Decryption methods. However, the qualities of ECC and LSSS access policies have been manipulated to allow for this increase in computational overhead to be non-existent or negligible at worst. This has resulted in a decreased total execution time while still providing the same level of fine grained access control.

This study suggests the Lightweight Multi-Authority KP-ABE Access Model (LMAKAM), to product a secure and computationally efficient security scheme. Multiple Attribute Authorities (AA) generate the public and private keys of the system through the use of ECC, while also defining the LSSS access policy of the system. The AAs utilize the access policy to determine how Data Owners (DO) encrypt data intended for Data Users (DU). Once the DOs encrypt the data, the AA generate the User Secret Keys (Usk) to be used by the DU in the decryption process. The Cloud Service Provider stores and performs a second encryption on the requested data before it is sent to the DU Assistant (DUA) for partial decryption. The DUA performs the majority of the computations required to decrypt the data, and then sends the partially decrypted data to the DU to complete the decryption process. The key contributions of this study are as follows:

- A lightweight but secure scheme has been presented using KP-ABE and ECC.
- The scheme utilizes LSSS in Key Generation in a manner that does not produce a large computational overhead.
- Outsourced decryption allows for low-computationally strong user-end devices in the IoT environment.
- The security analysis of the scheme is provided, as well as the performance efficiency when compared to a

prominent access control scheme in literature.

II. Preliminaries

TABLE I
TABLE OF NOTION USED IN SYSTEM

Notion	Description	Notion	Description
i	Private Key i	PK_i	Public Key i
α	Authorized Attribute Set	$(\mathbb{A}, \rho)$	Access Policy
csk, ck	Symmetric Keys	GID	General ID
R_u	Reconstruction Parameter	CA_{pvt}, CA_{pub}	CA Private and Public Key
$\mathbb{A}_x$	xth row of $\mathbb{A}$	Q_u, D_u,	DU Public and Private Key
Q_{DO}	DO Private Key	D_{DO}	DO Public Key
PK_α	Public Key for set α	$k_{i\in\alpha}$	Private Key for set α

A. Elliptic Curve Cryptography

For an Elliptic Curve *E* over F_q, where F_q is a prime finite field with integer modulo *q*, *E* is given by:

$$E : y^2 \text{ (mod q)} = x^3 + ax + b \text{ (mod q)} \tag{1}$$

where a,b,x and y are all elements of the curve *E*. It is also true that $4a^3 + 27b^2 \neq 0$. There is a cyclic group of order *r* known as *GR* which contains all the points on *E*. There also exists a point *G* called the generator point. All other points in the cyclic group *GR* can be produced by multiplying *G* by some integer value *i* as long as $i \leq r$. The number of subgroups present on the curve *E* can be determined to be the cofactor known as *h*. Elliptic Curves contain their own scalar multiplication, addition, inverse and null points (known as the point at infinity), which allow for cryptographic operations to occur.

In ECC, if two users wish to exchange a message the following is done. For simplicity, the two users will be refereed to as Device A and Device B or D_A and D_B. Both users need to produce their own respective public keys which are two points on the curve, say qk_A and qk_B, which are the scalar multiplications of the generator point *G* two integer values smaller than *q*, which are the the private keys of the users denoted as pk_A and pk_B. In other words:

$$qk_A = pk_A * G \tag{2}$$

$$qk_B = pk_B * G \tag{3}$$

If D_A wishes to send a message to D_B, then the plaintext message PT is then encrypted using the D_B's public key qk_B, and converted to the ciphertext CT. Then the receiver D_B can decrypt the CT using their private key pk_B.

B. Decisional Diffie–Hellman (DDH) Problem

The Decisional Diffie–Hellman (DDH) problem is a critical cryptographic assumption used to ensure security in various cryptographic systems. Consider a cyclic group G of prime order q with generator g. Let $a, b, c \in \mathbb{Z}_q$ be integers selected uniformly at random. The DDH problem involves distinguishing between the tuples (g^a, g^b, g^{ab}) and (g^a, g^b, g^c). The DDH assumption asserts that, given (g^a, g^b, g^z), it is computationally infeasible for an adversary to determine whether $z = ab \mod q$ or z is a random element in $\mathbb{Z}_q$.
In ECC, the DDH problem is similarly defined but within the context of elliptic curve groups. Let P be a generator point on an elliptic curve E over a finite field. For scalars $a, b, c \in \mathbb{Z}_r$ chosen uniformly at random, the DDH problem involves distinguishing between the tuples (P, aP, bP, abP) and (P, aP, bP, cP), where aP represents scalar multiplication of P by a.

C. Linear Secret Sharing Access Policies

Secret sharing is a method of dividing some secret s into shares. The secret can only be reconstructed if all the necessary shares of the secret are combined. In a LSSS Access Policy, shares are divided based on attributes. The boolean access tree is converted into what is known as a Sharing Matrix M and an attribute mapping ρ. For example, the access tree shown in Figure 1 can be converted to M and ρ also present in Figure 1 using the Lewko-Waters algorithm [18]. M is a *l x m*

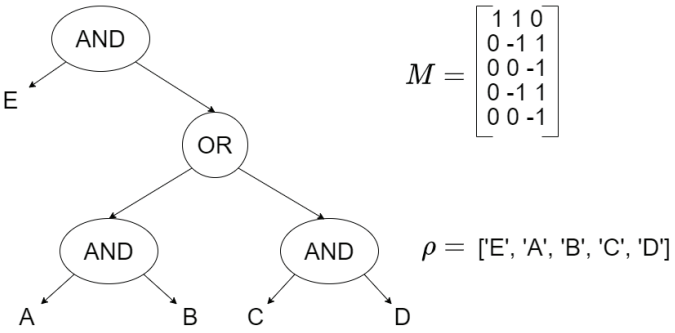

Fig. 1. Conversion of Access Tree to LSSS Access Policy

matrix where *l* is the is the total number of entities involved in the secret sharing scheme and *m* is the total number of attributes. To share a secret, a random vector v of length m is generated, where the first entry is s, the secret to be shared, and the rest are randomly generated numbers v_n. In other words $v = (s, v_1, v_2, ..., v_m)$. Multiplying the vector v by the rows of the matrix M based on the attributes of ρ gives λ_x, i.e. $\lambda_x = M_x \cdot v$, where M_x are the rows of the matrix mapped by ρ. For an authorized set α, a constant set c_x can be generated in polynomial time which satisfies the equation $\sum_{x\in[1,2,...,l]} c_x\lambda_x = s$. Using c_x, the secret can be reconstructed.

III. Overview of Proposed Scheme

A. System Model

There are eight entities in the access model, each with their own roles and responsibilities. The eight entities are:

- Data Owner (DO): Stores data from gateways locally. Performs encryption based on the attribute set obtained from the Attribute Authorities before data is sent to the Cloud Service Provider.

- Central Authority (CA): Generates the public parameters (PP) and registers each user, while maintaining a list of user details.
- Cloud Service Provider (CSP): Stores data from the system and plays a role in the encryption process.
- Data User (DU): Any device or entity hoping to access data in the system. DUs can perform decryption of data if they have the necessary attributes.
- DU Assistant (DUA): Performs partial decryption for DUs.
- Attribute Authority (AA):Generates the public and private keys for attributes in its domain, as well as defining the access policy to be used in the encryption process. The AA also is responsible for generating the user secret key based on the user's attributes.
- IoT Device: Connected to gateways and record data in the system.
- Gateways: Collect the recorded data of IoT devices and transfers the data to DOs.

The workflow of the access model can be seen in Figure 2. Here the interactions between different entities in the system can be seen.

IV. System Definitions

There are seven different algorithms used in the system to determine how data is exchanged in the system. These algorithms are as follows.

$System\_Setup(q \rightarrow PP)$: The CA defines the public parameters PP used by the system given any large prime number q. These PP are the characteristics of a Elliptic Curve which uses mod q.

$Authority\_Setup(PP \rightarrow (k_i, PK, (\mathbb{A}, \rho)))$: The AAs of the system generate the private keys k_i for each attribute in their domain, as well as their corresponding public key counter part PK given the PP. The AA also defines the access policy $(\mathbb{A}, \rho)$ based on the attributes it has authority over.

$Registration((R_u, ID) \rightarrow GID)$: The CA receives the registration request of a DU and generates a GID of the user based on the user's device ID and reconstruction parameter R_u. The user information is then stored on the CA.

$Encryption((PP, ck, (\mathbb{A}, \rho), \alpha, PT, PK, k_{i\in\alpha}) \rightarrow (CT, C_0, C_{1,x}, C_{2,x})$:The DO uses the symmetric key ck to encrypt the data that has been requested by the DU from the plaintext PT to the ciphertext CT. The DO then creates CT, C_0, $C_{1,x}$, $C_{2,x}$ using the access policy $(\mathbb{A}, \rho)$, the relevant public keys PK and the corresponding keys $k_{i\in\alpha}$ based on the attribute set of the DU α.

$Re - Encryption((CT, csk) \rightarrow CT_{CSP})$: The CSP re-encrypts the ciphertext CT using the symmetric key csk to create the new ciphertext CT_{CSP} .

$Key\_Gen((k_i, PK, \alpha) \rightarrow U_{sk})$: The AA generates the user secret key U_{sk} required to decrypt a message which has been encrypted for the attribute set α using the relevant private and public keys k_i and PK.

$Decryption((PP, CT_{CSP}, U_{sk}, csk, (\mathbb{A}, \rho)) \rightarrow (PT))$: The DUA and DU use the public parameters PP, user secret key U_{sk} and access policy $(\mathbb{A}, \rho)$, to decrypt the find the symmetric key ck and decrypt the ciphertext to the plaintext PT.

A. Design Goals

The proposed scheme is underpinned by three key design objectives, expanded upon as follows:

1) Security: In the contemporary landscape, there are a multitude of potential attackers who seek to compromise confidential of the data distributed in IoT environments. Consequently, ensuring the security of this data stands out as a paramount necessity.
2) Fine-Grained Access Control: Within an IoT environment, stringent control over access to data is imperative to prevent any unauthorized and illicit access attempts. The implementation of fine-grained access control, facilitated through an access policy, serves to either permit or deny access to specific data.
3) Lightweight Technique: Given the inherent resource constraints of most IoT devices, the design of an IoT environment must prioritize efficiency. This necessitates the development of a lightweight solution characterized by minimal computational costs and reduced power consumption.

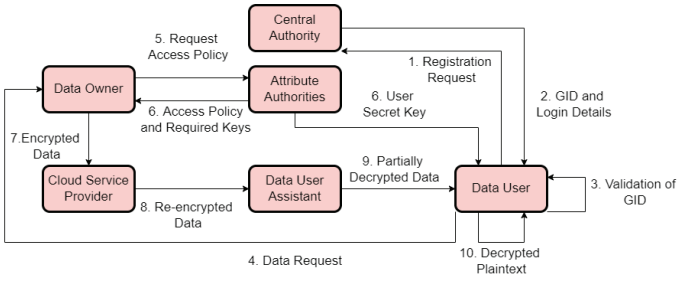

Fig. 2. Workflow of System

V. CONSTRUCTION

In transitioning from the CP-ABE system suggested in [17] to the KP-ABE system suggested here, the main changes occur in the Authority Setup (Algorithm 1), Encryption (Algorithm 3), Key Generation (Algorithm 2), Partial Decryption (Algorithm 4) and Final Decryption (Algorithm 5).

The main change is that the Access Policy is now defined by the Attribute Authority and not by the Data Owner. The Authority Setup can be seen in Algorithm 1, where the PP are used to create the secret keys k_i for each attribute in the domain as well as the corresponding PK for each attribute. The access policy is defined, where $\mathbb{A}$ is defined as the sharing matrix associated with the access structure and ρ is the appropriate attribute mapping in the form of $[A_1, A_2, ..., A_i]$ where A_i is the attribute defined by row i of the sharing matrix. This leads to a simpler encryption process for the

Data Owner, which can be seen in Algorithm 3. Here the DO uses the access policy obtained from the AA, the attribute set of the DU α, symmetric encryption, and encapsulation to encrypt the plaintext PT to ciphertext CT and obscure the encryption key ck in C_0. For the encapsulation process, the DO requires the summation of the public keys PK associated with the attributes of the attribute set PK_α and the corresponding private keys $k_{i\in\alpha}$. The CT is then encrypted again at the CSP

Algorithm 1 Authority Setup

PP i in Attributes $k_i \leftarrow k \in z_r$ $PK_i \leftarrow k_i \cdot G$ $\mathbb{A} \leftarrow M$ $\rho \leftarrow [A_1, A_2, ..., A_i]k_i, PK, (\mathbb{A}, \rho)$

and transformed into CT_{CSP} by using the symmetric key csk. In comparison with the CP-ABE system of [17], there is one less ECC multiplication that is required in the generation of $C_{1,x}$. Meaning per attribute, there is a significant decrease in computational complexity. The Key Generation Algorithm has also been changed by this process, which can be seen in Algorithm 2. Now, when the AA generates a data user's secret key based on the attributes of the user and their GID, there is no need for the second random integer y_i.

Algorithm 2 Key Generation

For the AA GID, α i in α $U_{sk_i} \leftarrow hash(GID) \cdot k_i$ $U_{sk}toDU$ For the DU $p \leftarrow p \in z_r$ i in α $U'_{sk_i} \leftarrow hash(GID) \cdot k_i + p$ U'_{sk} to DUA

Algorithm 3 Encryption

$PP, PT, ck, (\mathbb{A}, \rho), \alpha, PK_\alpha k_{i\in\alpha}$ $CT \leftarrow enc(PT)_{ck}$ $CT_H \leftarrow hash(CT) \cdot D_{DO} \cdot PK_\alpha$ define random vectors v and u and random secret s $\lambda_x = \mathbb{A}\cdot$ v $\omega_x = \mathbb{A}\cdot$ u $C_0 \leftarrow ck + (s \cdot PK_\alpha)_x$ $C_{1,x} \leftarrow \lambda_x \cdot PK_\alpha$ $C_{2,x} \leftarrow \omega_x \cdot PK_\alpha + k_{i\in\alpha} \cdot PK_\alpha C_0, (C_{1,x}, C_{2,x}), CT, CT_H$

This CT is then sent to the DUA for the first partial decryption, which can be seen in Algorithm Here the DUA uses the attribute set α of the DU. If this is an authorized attribute set, then the DUA can generate the vector c_x in polynomial time. Afterwards, the decryption process is completed by the DU using Algorithm 5 and the original PT is recovered. The PT is only accepted if the value CT_H can be recreated by the DU using the sum of the relevant public keys PK_α and the public key of the DO D_{DO}.
It should be noted that the calculation of D_x can also be defined by Equation 4, where the hash of the GID has been abbreviated to h_G.

$$D_x = \lambda_x \cdot PK_\alpha - h_G \cdot k_{i\in\alpha} - p \cdot PK_\alpha + h_G \cdot \omega_x \cdot PK_\alpha + h_G \cdot k_{i\in\alpha} \cdot PK_\alpha \tag{4}$$

This can be simplified further to Equation 5

$$D_x = \lambda_x \cdot PK_\alpha - p \cdot PK_\alpha + h_G \cdot \omega_x \cdot PK_\alpha \tag{5}$$

It can also be shown that N_1 can be defined by Equation 6

$$N_1 = \sum_{x\in X} c_x \lambda_{x\alpha} - \sum_{x\in X} c_x p_\alpha + \sum_{x\in X} c_x h_G \omega_{x\alpha} \tag{6}$$

This can further be simplified to the form found in Equation 7

$$N_1 =_\alpha \sum_{x\in X} c_x \lambda_x - p_\alpha \sum_{x\in X} c_x + h_{G\alpha} \sum_{x\in X} c_x \omega_x \tag{7}$$

Using the knowledge of linear secret sharing discussed in Section II., it can be seen that the term $\sum_{x\in X} c_x \lambda_x = s$ and $\sum_{x\in X} c_x \omega_x = 0$ yielding Equation 8.

$$N_1 = s \cdot_\alpha - p_\alpha \sum_{x\in X} c_x \tag{8}$$

This shows that the addition of N_1 and $p \cdot N_2$ is equivalent to $s \cdot PK_\alpha$. Therefore, the DU can obtain the original symmetric key ck.

Algorithm 4 Partial Decryption

$PP, CT_{csp}, csk, (\mathbb{A}, \rho), PK_\alpha$ $CT \leftarrow dec(CT_{csp})_{csk}$ $X \leftarrow \{x|\rho(x) \in \alpha\}$ $c_x \leftarrow c_x \in z_r$ $D_x \leftarrow C_{1,x} - U_{sk} \cdot PK_\alpha + hash(GID) \cdot C_{2,x}$ $N_1 \leftarrow \sum_{x\in X} c_x D_x$ $N_2 \leftarrow \sum_{x\in X} c_x PK_\alpha C_0, N_1, N_2, CT$

Algorithm 5 Final Decryption

$CT, CT_H, PP, N_1, N_2, C_0, p$ $ck' \leftarrow C_0 - (N_1 + p \cdot N_2)_x$ $hash(CT) \cdot Q_{DO} == CT_H$ $PT \leftarrow dec(CT)_{ck'}$ Request Data Again PT

VI. Security Analysis

The security of the proposed system is validated through correctness, resistance to attacks, and adherence to cryptographic assumptions. This analysis draws parallels to the security analysis presented in the work of Namasudra et al. [17], establishing that our improvements maintain the same level of security while adapting the algorithm for enhanced performance.

A. Security Under the DDH Assumption

Our system relies on the Decisional Diffie–Hellman (DDH) assumption to guarantee security, similar to the framework outlined by Namasudra et al. [17]. If the DDH assumption holds, the overall construction is secure.

Game Simulation: Let there be an adversary $\mathcal{A}$ with a non-negligible advantage $\epsilon > 0$. The adversary can query a secret key under the condition that the key cannot decrypt the challenge ciphertext (CT). This approach ensures that the security of a multi-authority system is equivalent to that of a single-authority system, as demonstrated in [17].

- **Initialize:** The adversary $\mathcal{A}$ selects an access policy and sends it to the challenger $\mathcal{C}$ as a challenge.
- **Setup:** The challenger $\mathcal{C}$ performs the Authority Setup, determines the public parameters (PP), and generates the private keys k_i and public keys PK.
- **Phase 1:** The adversary $\mathcal{A}$ submits a GID and attribute set α to the challenger $\mathcal{C}$ to obtain its private key. Since the private keys do not satisfy the access structure, $\mathcal{C}$ selects

a random scalar $p \in \mathbb{Z}_r$ and computes the secret key as $U'_{sk_i} = \text{hash(GID)} \cdot k_i + p$.

- **Challenge:** The adversary $\mathcal{A}$ selects two plaintexts PT_0 and PT_1 of equal length and submits them to $\mathcal{C}$. The challenger $\mathcal{C}$ randomly selects one of the plaintexts, calculates $C_0 = PT_c + s \cdot \text{PK}_\alpha$, and generates ciphertext components $C_{1,x}$ and $C_{2,x}$, which are sent back to $\mathcal{A}$.
- **Guess:** The adversary $\mathcal{A}$ outputs a bit $\mathcal{B}$, representing its guess for the chosen plaintext. If $\mathcal{B}$ matches the actual plaintext, $\mathcal{A}$ wins. The probability of $\mathcal{A}$ guessing correctly is $Pr(\mathcal{B} = \text{correct}) = \frac{1}{2} + \epsilon$, and the advantage of $\mathcal{A}$ is $\mathcal{C} = \frac{\epsilon}{2}$.

As with the scheme proposed in [17], the above analysis demonstrates that if the adversary's advantage is non-negligible, the challenger also gains a non-negligible advantage. Thus, the security of our scheme is upheld under the DDH assumption.

B. Correctness

Following the correctness principles outlined in [17], our scheme ensures the integrity of the decrypted data. Before performing the final decryption, the data user (DU) verifies correctness by checking $CT_H == \text{hash}(CT) \cdot Q_{DO}$. This guarantees that the original data is accurately retrieved.

C. Data Confidentiality

Similar to [17], our scheme ensures that users with insufficient attributes cannot obtain the necessary components N_1 and N_2 to reconstruct the symmetric key ck. Additionally, the partial secret keys U_{sk_i} are distributed across multiple attribute authorities (AAs), ensuring that no single entity (e.g., AA, CSP, or CA) can decrypt the data independently. The Delegated User Assistant (DUA) is also unable to fully decrypt the ciphertext as it does not possess the user's private key component p or GID.

D. Collusion Attacks

Our scheme inherits resistance to collusion attacks as demonstrated in [17]. The use of unique GIDs tied to the secret keys U_{sk} ensures that colluding users cannot combine their partial secret keys during decryption. Additionally, access to an authorized attribute set is required, eliminating the need for collusion to decrypt a message.

E. Key-Escrow Free

Building on the approach in [17], our system prevents key-escrow issues by distributing the key generation process across multiple attribute authorities. Even though the central authority (CA) oversees the system, decryption requires contributions from all AAs and the CSP, ensuring that no single entity can decrypt ciphertexts independently.

F. Man-in-the-Middle and Forgery Attacks

Following [17], our scheme protects against man-in-the-middle and forgery attacks through the use of the ciphertext hash CT_H. Adversaries are unable to forge CT_H without access to the data owner's (DO) private key D_{DO}. Thus, any tampering or interception of the ciphertext is detected, preserving the security of the system.

VII. Performance Analysis

This scheme was evaluated in terms of security provided and computational cost. It was compared to the scheme in [17] known as the LACMMAA, as well as two very powerful and well known schemes LEABE [19] and EBAC [20]. The algorithms explored were the Setup, Encryption, Pre-Decryption and Decryption Algorithms. Simulations were done for each algorithm, beginning with 10 attributes in the system and was increased in increments of 10 until 100 attributes.

A. Experimental Environment

This scheme was analyzed using Python and the tinyec library. For the scheme, the brainpoolP160r1 Elliptic Curve defined by: $y^2 = x^3 + 297190522446607939568481567949428902921613329152x + 173245649450172891208247283053495198538671808088$ mod 1332297598440044874827085558802491743757193798159 and a 160-bit key.

The experiments are conducted on a custom-built desktop system with the configuration of a Intel(R) Core(TM) i5-10400F CPU @ 2.90 GHz, 32 GB of RAM, 2-TB SSD and the Windows 10 Home Operating System.

B. Security Features

TABLE II
TABLE OF SECURITY FEATURES OF COMPARED SCHEMES

Scheme	ECC	Key-Escrow Free	Outsourced Decryption	Multiple Authorities
[17]	✓	✓	✓	✓
[19]	✓	×	×	×
[20]	✓	×	✓	✓
Proposed Scheme	✓	✓	✓	✓

The security features of this scheme have already been discussed, however, they are now compared to the other schemes in Table II. It can be seen that this scheme provides the same level of security as the LACMMAA [17] scheme as they have similar constructions. This scheme also has an advantage over the [19] as it provides protection against key-escrow, has multiple authorities and outsources decryption. Similarly, it has an advantage over EBAC [20] as it provides protection against key-escrow.

C. Computational Cost

The performance of the system is assessed through one crucial metric, computational cost in milliseconds. The execution time of an algorithm can be measured by the amount of ECC scalar multiplications required as attributes increase. This is because a single scalar multiplication, represented by $\mathbb{M}$, is the most computationally intense procedure performed by the algorithms. The execution time in terms of $\mathbb{M}$ can be seen in Table III.Here it can be seen that T_A, T_L and T_R represent the total number of attributes in the access structure, the total number of leaf nodes in the access structure and the total number of attributes of the receiver respectively. It should be noted, that T_L and T_R have non-linear relationships to T_A. T_L and T_R depend on the defined Access Policy, but these variables can be expected to increase as T_A increases, as they are directly proportionate.

These execution times can be found by finding the amount of $\mathbb{M}$ operations performed in each algorithm. In the setup portion, the CP-ABE scheme required two $\mathbb{M}$ operations, namely $y_i \cdot G$ and $k_i \cdot G$, for each attribute in the system, while our definition only requires one for $k_i \cdot G$. However, our scheme does require the this algorithm to also generate the Access Policy which will add to the delay. Similarly, the removal of the $y_i \cdot G$ from the encryption stage signifies the scheme proposed here takes one less $\mathbb{M}$ operation per attribute considered.

It can be seen that the proposed scheme matches the Setup execution time of the best performing scheme, in this case LEABE [19], while out performing the LACMMAA [17]. However, since this scheme utilizes multiple AA while LEABE does not, this means that this execution time while be distributed amongst the authorities and operate slightly more efficiently. The proposed scheme outperforms LACMMAAA [17].

This is due to the fact that this scheme was made to be an improvement over LACMMAA, it removed the need for one scalar multiplication and therefore should be comparatively faster as the number of attributes in the system increases. Since this scheme managed to maintain the same decryption algorithm as LACMMAA, it benefits from having a decryption time of $(1)\mathbb{M}$. This makes the proposed scheme ideal for IoT environments with resource constrained user-end devices.

This should also signify that the scheme will have a comparable total execution time to the compared schemes while still providing the robust security of LACMMAA.

TABLE III
TABLE OF EXECUTION TIME OF ALGORITHMS OF COMPARED SCHEMES

Scheme	Setup	Encryption	Pre-Decryption	Decryption
[17]	$(2T_A+1)\mathbb{M}$	$(3T_L+1)\mathbb{M}$	$(T_R)\mathbb{M}$	$(1)\mathbb{M}$
[19]	$(T_A+1)\mathbb{M}$	$(T_L+1)\mathbb{M}$		$(T_R+1)\mathbb{M}$
[20]		$(T_L+1)\mathbb{M}$	$(T_R+1)\mathbb{M}$	$(3)\mathbb{M}$
Proposed Scheme	$(T_A+1)\mathbb{M}$	$(2T_L+1)\mathbb{M}$	$(T_R)\mathbb{M}$	$(1)\mathbb{M}$

In Figures 3, 4, 5, and 6, the execution time of the Key Generation, Encryption, Decryption and Total Execution Time of the proposed scheme versus those of LACMMAA [17], LEABE [19] and EBAC [20] can be seen. It can be seen that the proposed scheme performs better than LEABE and EBAC for Key Generation, showing improvements of up to 75ms and 100ms in execution time, and is only marginally slower, at worst 20ms slower, than LACMMAA. This is due to the low level of complexity in the generation of the U_{sk} and only having to perform one scalar multiplication per attribute. It is slightly slower than LACMMAA due to the fact that the Key Generation process also requires the AA to generate the access policy. The Encryption algorithm performs better than

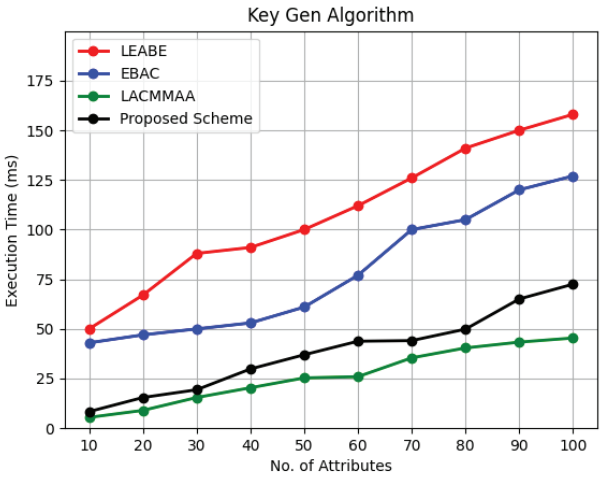

Fig. 3. Execution Time of Key Generation Algorithm

LACMMAA, up to 100ms quicker execution time, due to the fact it has one less scalar multiplication per access tree leaf but does not perform as well as the EBAC and LEABE, being at worst 250ms and 75ms slower in exectuion time, due to the fact it utilizes LSSS instead of a more simple access structure such as an AND Gate or Threshold structure. The system

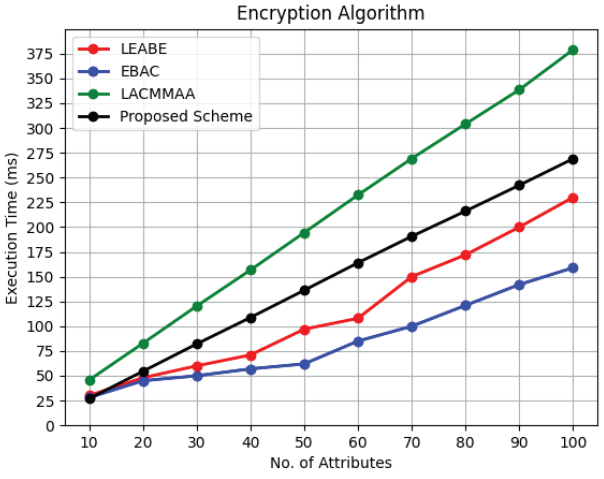

Fig. 4. Execution Time of Encryption Algorithm

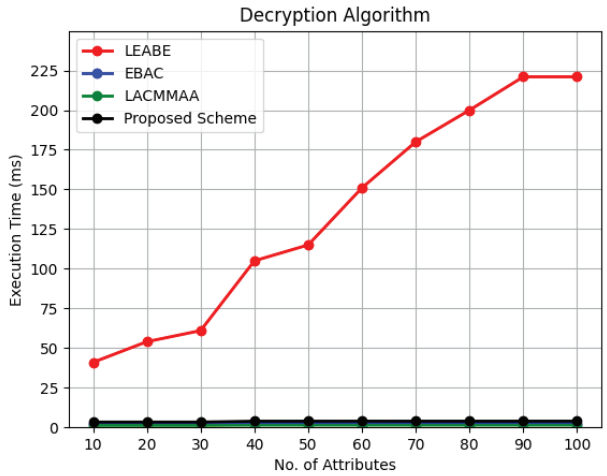

Fig. 5. Execution Time of Decryption Algorithm

performs as well as EBAC and LACMMAA in the execution time of the Decryption Algorithm, with the differences between the three being negligible and due to chance. It vastly outperforms LEABE, which shows the slowest execution time for Decryption, by up to 225ms. As is apparent in Figure 6, the proposed scheme outperforms LEABE and LACMMAA, having total execution time improvements of up to 250ms and 125ms respectively. The differences in execution time between the proposed scheme and EBAC are negligible, on average the discrepancy is within 12.5ms. Therefore, it can be concluded that the scheme is the second most efficient of the four while providing robust, fine-grain security. This is due to the fact that the system is built upon the principles that made LACMMAA so robust, while incorporating a KP-ABE structure to allow for faster Encryption times.

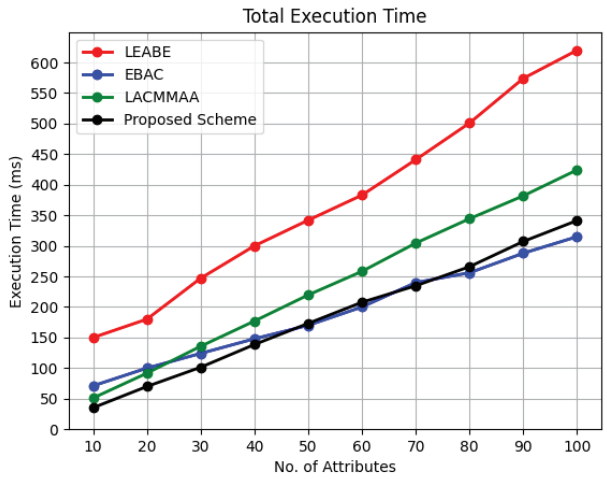

Fig. 6. Total Execution Time of All Algorithms

VIII. Conclusion

In this study, a lightweight, fine-grain access model was created utilizing ECC and KP-ABE. It provides robust security against attacks such as collusion attacks, man-in-the-middle attacks and forgery attacks. This scheme provides data confidentiality and ensures data correctness. The use of multiple AAs allows this scheme to be secure against key escrow. The scheme also allows for outsourcing decryption, making it ideal for IoT environments that have resource constrained user-end devices.

The scheme was compared to popular lightweight schemes found in literature, and it was found that the scheme provided robust security while still having better execution times than the selected schemes. The scheme may benefit from the use of edge-computing to allow better security against key escrow and attacks such as double-spending and record hacking. The issue of attribute revocation could also be addressed through the use of edge-computing.

References

[1] M. De Donno et al. ”Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog,” in IEEE Access, vol. 7
[2] A. Subrahmannian and S. K. Behera, ”Chipless RFID Sensors for IoT-Based Healthcare Applications: A Review of State of the Art,” in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-20, 2022
[3] O. Elijah et al., ”An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges,” in IEEE Internet of Things Journal, vol. 5, no. 5,
[4] M. Asplund and S. Nadjm-Tehrani, ”Attitudes and Perceptions of IoT Security in Critical Societal Services,” in IEEE Access, vol. 4
[5] M. Yahuza et al., ”Systematic Review on Security and Privacy Requirements in Edge Computing: State of the Art and Future Research Opportunities,” in IEEE Access, vol. 8
[6] M. Alrowaily and Z. Lu, ”Secure Edge Computing in IoT Systems: Review and Case Studies,” 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, 2018
[7] K.A. Shim, ”A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks,” in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 577-601
[8] C. Meshram et al., ”A Lightweight Provably Secure Digital Short-Signature Technique Using Extended Chaotic Maps for Human-Centered IoT Systems,” in IEEE Systems Journal, vol. 15, no. 4, pp. 5507-5515, Dec. 2021
[9] P. Liu, ”Public-Key Encryption Secure Against Related Randomness Attacks for Improved End-to-End Security of Cloud/Edge Computing,” in IEEE Access, vol. 8, pp. 16750-16759, 2020,
[10] C. Xiyuan, W. Di, L. Jian and Z. Miaoliang, ”A Security Violation Detection Method for RBAC Based Interoperation,” 2006 International Conference on Computational Intelligence and Security, Guangzhou, China, 2006, pp. 1491-1496, doi: 10.1109/ICCIAS.2006.295308.
[11] L. Hong-Yue, D. Miao-Lei and Y. Wei-Dong, ”A Context-Aware Fine-Grained Access Control Model,” 2012 International Conference on Computer Science and Service System, Nanjing, China, 2012, pp. 1099-1102, doi: 10.1109/CSSS.2012.278.
[12] I. Huso et al., ”Distributed and Privacy-Preserving Data Dissemination at the Network Edge via Attribute-Based Searchable Encryption,” 2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet), Pafos, Cyprus, 2022, pp. 122-130
[13] Z. Wan, J. Liu and R. H. Deng, ”HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing,” in IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 743-754, April 2012, doi: 10.1109/TIFS.2011.2172209.
[14] X. Li, T. Liu, C. Chen, Q. Cheng, X. Zhang and N. Kumar, ”A Lightweight and Verifiable Access Control Scheme With Constant Size Ciphertext in Edge-Computing-Assisted IoT,” in IEEE Internet of Things Journal, vol. 9, no. 19, pp. 19227-19237, 1 Oct.1, 2022, doi: 10.1109/JIOT.2022.3165576.
[15] J. Li et al., ”Securely Outsourcing Attribute-Based Encryption with Checkability,” in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 8, pp. 2201-2210, Aug. 2014,
[16] V. Odelu & A.K. Das, “Design of a new CP-ABE with constant-size secret keys for lightweight devices using elliptic curve cryptography.” Security and Communication Networks., 2016, 9. 10.1002/sec.1587.

[17] S. Das and S. Namasudra, "Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure," in IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 821-829, Jan. 2023, doi: 10.1109/TII.2022.3167842.
[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.
[19] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," Future Gener. Comput. Syst., vol. 49, pp. 104–112, 2015.
[20] X. Qin, Y. Huang, and X. Li, "An ECC-based access control scheme with lightweight decryption and conditional authentication for data sharing in vehicular networks," Soft Comput., vol. 24, pp. 18881–18891, 2020.
[21] J. Ma, J. Liu, X. Huang, Y. Xiang and W. Wu, "Authenticated Data Redaction with Fine-Grained Control," in IEEE Transactions on Emerging Topics in Computing, vol. 8, no. 2, pp. 291-302, 1 April-June 2020, doi: 10.1109/TETC.2017.2754646.
[22] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[23] Hu, S., Zhong, T., He, H. et al. "Attribute-based encryption of LSSS access structure with expressive dynamic attributes based on consortium blockchain." Ann. Telecommun. 78, 509–524 (2023).
[24] D. Hankerson, A. Menezes, and S. Vanstone, "Guide to Elliptic Curve Cryptography," Springer Science & Business Media, 2004.
[25] D. Huang, Q. Dong, and Y. Zhu, "Attribute-Based Encryption and Access Control (Data-Enabled Engineering)," CRC Press, 2020.

Agustin Ferrer-Rojas was born in Venda, Limpopo, South Africa in 2000. He received a B.Eng in Electronic engineering from the University of Pretoria in 2024.

BODHASWAR T. MAHARAJ received his Ph.D. in engineering in the area of Wireless communications from the University of Pretoria. Dr Maharaj is a full professor and currently holds the research position of Sentech Chair in Broadband Wireless Multimedia Communications (BWMC) in the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria. His research interests are in OFDM-MIMO systems,massive MIMO, cognitive radio resource allocation and 5G Cognitive Radio Sensor Networks.

CONTENTS PAGE