ATTRIBUTE BASED APPROACHES FOR SECURING DATA SHARING IN INDUSTRIAL CONTEXTS

Mohd Sajid[1], Md. Ateeq Ur Rahman[2],Sridhar Gummalla[3]
[1]PG Scholar, Department of IT, Shadan College of Engineering and Technology, Hyderabad, Telangana -500086
sajid495aleem@gmail.com
[2]Professor, Department of CSE, Shadan College of Engineering and Technology, Hyderabad, Telangana -500086
mail_to_ateeq@yahoo.com
[3]Professor, Department of CSE, Shadan College of Engineering and Technology, Hyderabad, Telangana-500086
Sridhar_gummalla@yahoo.com

ABSTRACT

Data sharing is becoming increasingly crucial in the manufacturing and process industries, which use data-driven models and advanced analytics to estimate wear and tear and analyze production performance. Policies in these types of situations must be simple to implement, and data access must be rigorously regulated to prevent leaks to unauthorized individuals. Furthermore, sharing data with users outside of trusted domains should be done using encryption. Finally, there must be methods for withdrawing access to data. This paper provides an overview of attribute-based data access control strategies, focusing on policy administration and enforcement. Our goal is to identify essential functions that Attribute-Based Access Control (ABAC) and Attribute-Based Encryption (ABE) offer that can be coupled to achieve the requirements. We discuss these potential combinations in the context of a recommended architecture for secure data exchange. The report finishes by identifying knowledge gaps that will assist guide future research on attribute-based techniques for safe data transmission in industrial environments.

Keywords: Attribute-Based Access Control(ABAC), Attribute-based encryption and decryption for secure data.

1. INTRODUCTION

Since data fosters innovation, value creation, new services, and process improvement, it is essential to the Industry 4.0 revolution [1]. The sources of the data can include user or corporate data that is completely confidential, financial accounts, sensor data, and more. Companies and sectors often work together with other organizations to fulfill customer demands, adhere to regulatory standards, or create superior products. These kinds of alliances and partnerships create ecosystems in which data is often the primary resource. Therefore, data needs to be shared among organizations within a data-driven ecosystem in order to be leveraged as a strategic resource for generating targeted values, innovations, or process improvements.

Conflict arises when data is used as a strategic resource in ecosystems where value creation depends on cooperation. Given its inherent importance, there is a need to safeguard data in addition to ensuring that it is shared efficiently. Furthermore, because of its significance in a particular process or the potential insights it may offer, certain data may be intrinsically more valuable than others. Additionally, certain data may need to be kept private in order to preserve business-critical information or personal data that is required to be protected by the EU's General Data Protection Regulation (GDPR) [3]. Other data may need to be made publicly accessible.

Currently, third-party cloud servers and services like Microsoft's Azure [4] and Amazon's S3 [5] have grown in popularity in addition to corporate data centers, as part of the efforts to enable data-driven ecosystems and driven by needs like high availability and extendable capacity. This lowers the cost of owning and maintaining infrastructure and permits increased collaboration, agility, and scalability without requiring a significant upfront investment. However, because data are now not under the owner's control, using third-party cloud servers brings up additional security issues. A lot of potential adopters, such industries or organizations, are discouraged by this fear. Therefore, it is essential to guarantee the integrity, privacy, and confidentiality of the outsourced data. [6], [7], and [8]. It may also be necessary to take organizational and legal needs into account [9].

Different levels of data protection are required because, although overly lax enforcement could result in sensitive data leaks and negatively impact the company, extremely tight laws surrounding public data sharing might needlessly hold down the process [2]. Role-Based Access Control (RBAC) [10] or the more contemporary idea of Attribute-Based Access Control (ABAC) [11], [12] can be used to safeguard data based on policies. The purpose of ABAC is to determine and enforce a set of permitted activities by the subject upon the object by assembling policy, subject attributes, and object attributes [13]. Moreover, the permitted operations may be enforced by means of environmental circumstances [14].

Inside a trusted domain, where access control policies may be applied without any sensitive data leaving the secure realm, access control techniques based on RBAC or ABAC can regulate the flow of data while it is being exchanged. Nevertheless, those access

control measures might not be adequate if third-party data storage service providers aren't entirely reliable. Furthermore, when data is sent between a source or storage and a user outside of a trusted domain, it could leak. Therefore, data leakage in untrusted domains and while traveling over public networks cannot be stopped by access control alone.

Encrypting the data is a traditional method for handling cloud environments and third-party service providers that are not entirely trustworthy. One can employ standard encryption techniques like Public Key Infrastructure (PKI) [15]. Pretty Good Privacy is one such strategy (PGP). PGP is extensively used for jobs like messages, emails, and file sharing apps. It uses symmetric key and public key encryption algorithms to ensure data privacy when sending and receiving data. When preparing data for transmission, the standard procedure is to encrypt the message using a session key a random key—to produce a ciphertext. Next, the session key encrypted with the recipient's public key is bundled with the ciphertext. The package is then delivered to the final user, who uses its own private key to get the session key and, ultimately, the message that lies behind it [16]. But because the encryption work is done during the transmission step, this kind of technique is expensive at runtime. As an alternative, several encrypted versions of the data might be kept for various users with various public keys. However, this raises the cloud's storage overhead, particularly when there are a lot of users.

Attribute-Based Encryption (ABE) solves the runtime cost and storage problems associated with per-end-user encryption. Because ABE enables attribute-based encryption, a data owner need only predefine sufficient attributes to satisfy the desired level of access granularity, negating the need to consider the system's user count. Furthermore, the encryption process can be separated from the transmission phase by enabling the data owner to encrypt the data in the first place using only their attributes—that is, their public keys—instead of having access to the end users' full information [17]. ABE and ABAC share similar properties, so that ABE solutions naturally fit into the ABAC idea because of the nature of their attributes [14].

Sectors generally store data on cloud platforms as well as in corporate data centers. For the data to be securely protected, encryption and access control are therefore required. This begs the question of how to combine these complementary mechanisms. The topic of how to use ABE to achieve ABAC has been examined in relation to ABE-enabled ABAC [14]. Further research is necessary, nevertheless, on the use of ABAC in trusted domains in conjunction with ABE when ABE is not employed. However, research has been done on access control models with ABE enabled.

In this work, we examine how attribute-based methods, such as ABAC, may be used to restrict access to data in trusted domains. We also explore how these control concepts might be extended to mobile enforcement of access controls in untrusted domains using ABE. In order to determine the respective justifications and State-of-The-Art (SoTA) in light of these qualities, we identify critical properties for safe data sharing in untrusted domains and survey ABAC and ABE. In order to investigate potential ways to integrate these attribute-based approaches into a cohesive system for safe data sharing that supports both trustworthy and untrusted domains, we pay particular attention to architectural elements, concepts, and rules of usage. We also investigate the SoTA approaches for merging ABE and access control paradigms, which mostly use ABE-enabled access control models rather than RBAC models as previously noted.

2. RELATED WORK

L. Hui, C. Chunjie, and S. Jingzhang In a cloud medical context, searchable encryption technique based on CPABE with attribute updating [2018]. Hospitals are saving money by outsourcing their encrypted electronic medical records to cloud services thanks to advancements in cloud storage. The searchable encryption systems that are now in use are unable to handle both fine-grained access control and ciphertext search in a cloud medical context where the attribute is changed often. Thus, a cryptographic retrieval technique allowing attribute update is presented by merging searchable encryption technology with ciphertext policy attribute-based encryption. The cloud storage server receives a partial decryption transmission and allows for frequent attribute updates.Under the DBDH assumption, security analysis demonstrates that the scheme can ensure security and privacy, and real-world testing findings demonstrate the system's effectiveness and usefulness.

H. Yin, J. Zhang, Y. Xiong, L. Ou, F. Li, S. Liao, et al., ABSE: A searchable encryption scheme based on ciphertext policies and attributes[2019]. An efficient method for achieving safe search over encrypted data is searchable encryption. One common use case for searchable encryption is when a data owner uploads encrypted data to a server, and the server uses a query trapdoor that a data user submits to efficiently conduct keyword-based search over the encrypted data. Both the user's and the owner's queries are kept private on the server. In an effort to improve security and performance, secure data updateability (dynamics), and search results verifiability, numerous searchable encryptions have recently been developed. Unfortunately, the majority of currently available works grant the data user infinite search capabilities without considering their search permissions. In actual use,

giving data users the ability to search the data is a crucial step in enforcing data access control. In this research, we leverage the ciphertext-policy attribute-based encryption technique to present an attribute-based searchable encryption system. With our system, a data user can request a fine-grained search authorization from the data owner. The basic concept is that the data owner under a designated access policy encrypts an index keyword, and the data user can only search over the encrypted index keyword if and only if the data user's attributes meet the access policy requirements. For our scheme, we offer comprehensive studies of correctness, performance, and security. The comprehensive tests show that, in many respects, our suggested method works better than Zheng's comparable approach, CP-ABKS.

Y. Zhang, Q. Yu, and J. Li Encryption based on hierarchical attributes and continuously leakage-resistant [2019]. Because of its fine-grained access control, attribute-based encryption, or ABE, is commonly used in cloud computing environments. The majority of ABE schemes overlook side channel attacks, which have the potential to reveal cryptosystem secrets. The goal of leakage-resilient cryptography is to simulate protection against different side channel attacks. The formal definition and security model of hierarchical attribute-based encryption (HABE) with continuous leakage-resilience are presented first in this article. Additionally, we provide a continuous leakage-resilient ciphertext-policy HABE method. Both master key leakage and secret key leakage are resistant to the suggested scheme. We use dual system encryption techniques to demonstrate the security of our approach under composite order bilinear group assumptions. A theoretical analysis is conducted on leakage-resilience performance. The relative leakage ratio is nearly up to a third if our suggested strategy is implemented at that depth. Furthermore, we provide a performance comparison using experiments.

3. METHODOLOGIES

Key Properties of Secure Data Sharing:

In a trusted and controlled domain, access control systems like RBAC or ABAC are effective at preventing unwanted access to data. However, due to the lack of complete data security management for the data owners, cloud-based services are untrusted or vulnerable to sincere but suspicious attacks, rendering standard access control methods inadequate to protect data privacy [7], [23].

In a range of applications, cryptographic techniques like ABE have shown to be effective solutions while handling outsourced data. Using attribute-based policies, ABE converts the stored data into ciphertexts that can only be decrypted by users who have the right attributes. There are certain difficulties with using ABE techniques to safeguard industrial data, nevertheless. Since ABE does not employ standard policy language like that provided by the ABAC standards, creating and managing policies in particular is a significant problem [24]. This encourages us to look into the best way to integrate ABAC with ABE. We establish a set of critical qualities for this inquiry that are inefficiently covered by either ABE or ABAC alone. We choose characteristics that are crucial for a safe data sharing plan that uses cloud computing and external storage for industrial data. We use these qualities to explore the State-of-The-Art (SoTA) for ABAC and ABE in Sections III and IV. In Section V-C, we further utilize the attributes to assess the combined scheme for ABAC-enabled ABE.

Key properties for secure data sharing:

Fine-granularity: The precise files, database tables, or sensor measurements that the access control policies are applied to can all be considered at different granularities. An essential feature is the capacity to effectively administer and implement fine-grained regulations, such as those based on particular values in time-series data. The expressiveness of the policy language and the simplicity of managing attributes are two more properties that affect how granular the access policies can be. These properties will be covered in the next section.

Expressiveness: The ease of articulating policies for a policy language determines this feature. While some policy languages might need completely defined elements or the development of additional elements, others might offer beneficial abstraction. It is crucial to remember that two policy languages do not always have to be equally expressive in order to describe the same kind of policies. We regard policy language B to be more expressive than policy language A if policy language A requires more work to communicate the same policy.

Dynamics: IoT ecosystems for industry are dynamic. The network is always being joined and joined by new sensors and users. After the system is initialized, access control schemes may or may not be intended to establish or alter access policies. Furthermore, it could be necessary to restart the system or re-encrypt the ciphertext when access is revoked.

Ease of management: One of the main things that prevents the implementation of fine-grained access control is a heavy administrative burden. This kind of policy's development could result in a lot of characteristics that are challenging to effectively control. Certain technologies could be useful, like automatic attribute assignments or hierarchical attribute management.

Access revocation: Efficient access revocation capabilities are necessary in industrial data sharing applications, particularly in cases where collaboration with a partner ends. Access revocation may necessitate ciphertext re-encryption, keying material redistribution, or central authority intervention, depending on the method selected.

Implementation/Deployment: Even if an industry's access control system is inefficient or not completely safe, it is frequently already in place. One of the reasons attribute-based techniques have not become industry standards is the large amount of work involved in their first deployment. Then, solutions are required, such as compatibility with current standards or even tools to migrate from already-existing access control systems.

ATTRIBUTE-BASED ACCESS CONTROL APPROACHES FOR SECURE DATA SHARING

A. Background

1) Mandatory Access Control **(MAC)**

The system administrator assigns users clearance levels and confidentiality levels to data while implementing Mandatory Access Control (MAC). The security policy administrator has complete control over all operations pertaining to the relationship between the data labels and the user level. This implies that individual data owners in a pure MAC system are not directly in charge of the rules governing their shared data. For institutions like the military or governments, MAC policies are helpful in establishing policies that apply to the entire organization. Strict sensitive data privacy standards can be enforced while allowing information sharing between businesses by combining MAC regulations with other strategies like Discretionary Access Control (DAC) [25].

2) Discretionary access control **(DAC)**

Discretionary access control, first proposed in the early 1970s, is a way to limit access to resources according on the user's identity. A resource in DAC is usually under the control of its owner. The Lampson access matrix approach, which assigns resources as columns and subjects as rows, serves as the foundation for access control. The subject's access privileges over the resource are represented by the entry in the matrix under the subject and resource. Because the owner or subjects with the appropriate access rights can, at their discretion, provide such permissions to any other user, access control is seen as discretionary [26].

3) RBAC

Role-Based Access Control is one method that can implement both DAC and MAC policies (RBAC). The foundation of RBAC is the notion that, in many companies, decisions about access control are made in response to the roles of the individuals making data requests. Within an organization, roles act as a group where individuals share a set of access permissions to resources. Roles are closely tied to job functions. Since a role's duties are more reliable than a user's, it makes sense from a management perspective to grant access privileges to roles rather than to specific users. On the other hand, users' membership in roles tends to vary over time. As a result, RBAC makes security administration easier and makes giving and removing access rights simpler.

In addition, RBAC presents the idea of role hierarchy as a way to match roles to levels of authority and responsibility in an organizational chart. Senior jobs—those at the top of an organizational chart—generally have greater authority than junior roles, which are at the bottom. In actuality, the top-level jobs ought to inherit any additional permissions granted to them along with the access privileges from those beneath them [10].

Although RBAC is helpful and easy to manage in small, static companies, it can be difficult to manage an RBAC scheme in larger, more diverse organizations. When the number of users and roles grows out of control, problems known as role explosion may arise. This is typical of large businesses where a large number of specialized workers carry out a variety of specialized jobs. In order to attain fine granularity in big diverse organizations, research has been performed to build more flexible and manageable access control systems as a result of these scalability concerns [27].

B. Key Concepts and Mechanisms

The access control paradigm known as attribute-based access control, or ABAC, bases its definition of policies over individuals doing actions on objects on given attributes. This paradigm functions as a substitute for conventional access control approaches, including Mandatory Access Control (MAC), Role-Based Access Control (RBAC), and Discretionary Access Control (DAC). ABAC builds access policies utilizing attributes rather than the user identity as the foundation for access control. These characteristics fall into one of the five groups listed below:

The qualities of topics within the system are described by user attributes. These characteristics may include the name, age, clearance level, work title, and so on.

The characteristics of the resources in the systems are described by object attributes. These may consist of the author, security level, file type, format, and so forth.

The system's current state is described by its environmental attributes, which include things like the day of the week, the time of day, the quantity of users, etc.

Connection attributes describe the current session of the user, such as the location, IP address, session start date and time, etc.

Administrative attributes define the system's configuration properties, including the minimum level of trust needed to gain access or the maximum session duration.

Fundamentally, ABAC access policies leverage the outcome of Boolean operations between attributes to restrict user access to resources or objects. Objects (O) are the set of protected objects in the system; Attributes (A) are the set of possible actions to perform over objects that can be authorized to users; Permissions (PERM) are the set of possible actions to perform over objects that can be authorized to users; and finally, Policies (P) are the set of policies governing access to the system. These elements are the most frequently found in ABAC systems [28].

Relations in the ABAC model connect all of these elements; specifically, resources are assigned attributes by Object Attribute Assignments (OOAs), users are assigned attributes by User Attribute Assignments (UAAs), and policies are connected to the access permissions that have been given by Policy Permission Relations (PRRs). Policy languages have an impact on how these interactions and assignments are implemented, particularly on how policies are granted permissions.

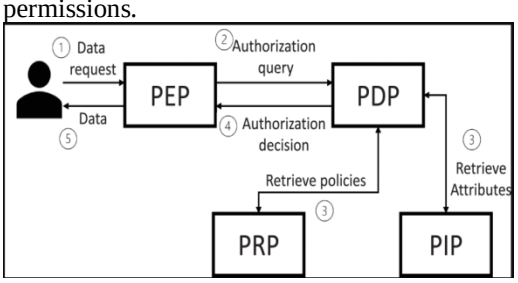

Fig 1. Basic ABAC architecture.

ABAC architectures, like the one shown in Figure 1, typically have a minimum of one policy enforcement point that handles requests for user data. The request is forwarded by this PEP to a Policy Decision Point (PDP), as shown in Figure 2, which obtains pertinent attributes from the Policy Information Point (PIP) and pertinent policies from the Policy Retrieval Point (PRP). After assessing the policies and characteristics, the PDP responds to the first request by generating an authorization decision that is forwarded back to the PEP. The user does not receive the requested data from the PEP until the authorization decision is approved.

Figure 2,The permission decision-making process might incorporate additional elements, such conditional or environmental features. Furthermore, given these kinds of criteria, various PEPs implementing policies in various locations may result in various authorization decisions [9]. These characteristics can be used in industrial settings to separate particular kinds of data according to the kind of processes carried out there.

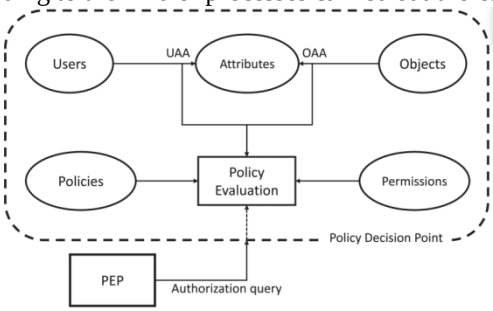

Fig 2. ABAC Policy Decision Point Internal Flow.

4. OUR PROPOSED MODEL

Attribute-Based Encryption:

A. Background

ABE is a public key encryption method intended for group decryption as opposed to individual user use. This method replaced the earlier Identity-Based Encryption (IBE) systems, in which the public key was typically a string that could be used to uniquely identify a single person, like their email address or social security number. By using the recipient's identity to encrypt the message, IBE schemes enable owners to transmit confidential messages to other users without having access to their public keys. The underlying communication is only accessible to users whose identities match those in the ciphertext.

As a component of a fuzzy identity-based encryption technique, attribute-based encryption was initially presented by. In this scheme, a set of qualities that are utilized to encrypt a message explain the identity. The message can only be decrypted and obtained by users whose given attributes coincide with those that were used to produce the ciphertext. If the sets match by a predetermined threshold k, the attributes overlap. This indicates that in order to decode the ciphertext, the user needs to possess at least k attributes from the ciphertext attribute set.

In contrast to conventional encryption systems, the use of attributes when defining keying material while implementing ABE in industrial data sharing contexts lowers the computational and administrative cost linked to encryption. Rather than encrypting every time a message is transmitted, ABE enables data to be encrypted once and distributed to several separate receivers. Furthermore, defining the keying material in terms of qualities makes it possible to create ciphertexts without completely defining the end users, which streamlines the setup procedure.

B. Key Concepts and Mechanisms

In this section, we describe the basic components of ABE.

1) Access Structure and Access Trees

The owner of a secret typically wishes to share it with a select group of people in a secret sharing arrangement. The access structure is the name given to this group. Within an encryption scheme, the parties that may decrypt a communication are represented by the access structure; persons outside of this structure are unable to divulge any information regarding the shared secret. The set of permitted characteristics is contained in the access structure, specifically for attribute-based encryption.

Access trees are a useful representation for access structures, in which each nonleaf node and an attribute represent a threshold gate by each leaf node. When threshold gates are present in nonleaf nodes, they can be represented by logical AND gates if the threshold is equal to the number of child nodes connected to the gate (n -of-n) or by logical OR gates if the threshold is 1 (1-of-n).

In ABE systems, monotone access structures are the most often utilized access structures. Any set that has the set of permitted attributes can meet the access structure in order for it to be considered monotone. Stated differently, every superset of B must be able to reconstruct the secret if a subset of characteristics B can do so [38]. One drawback of employing monotone access structures is that they do not enable logical NOT gates, meaning that negative attributes cannot be employed effectively while building the access tree. Still, by treating the not-attribute as a stand-alone attribute in the tree, negative attributes can be inefficiently expressed in monotone access structures.

2) Basic Construction Algorithms

Attribute-based encryption schemes follow a basic construction structure based on the following 4 primary algorithms: setup, encryption, key generation, and decryption.

- **Setup:** The algorithm that takes any implicit security parameter and creates public parameters and a master key. In this step, the universe of attributes is defined.
- **Encryption:** The algorithm that applies encryption to a message.
- **Key generation:** The algorithm that generates decryption keys based on a set of attributes.
- **Decryption:** The algorithm that decrypts a ciphertext to obtain the underlying message.

The relationship between the access structure and the user attributes is established by the ABE scheme that is in use. This is the primary distinction between the Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE), the two basic ABE methods. The user's private key is linked to the access policy in KP-ABE, and the ciphertexts are tagged with a set of attributes. Therefore, in this approach, the key generation method handles the access structure that determines whether or not a user is permitted to decrypt a ciphertext. In contrast, CP-ABE encrypts the ciphertexts using an access policy and links the user's private key to a collection of attributes. The encryption method in CP-ABE manages the access structure, producing a ciphertext that can only be decrypted by a limited group of people who meet certain requirements.

- **ABAC-Enabled Abe**

A. Rationale

ABE is a promising solution for access control in a cloud environment. However, it has some implementation issues, specifically in attribute policy construction. ABE policies are not usually designed to be created from an RBAC or ABAC base. Moreover, ABE data policy definition and expression do not follow any standard language. The methods used to build ABE policies, such as access structures, lack flexibility and expressiveness.

ABE schemes rely on monotonic access structures to define and express access control policies. These monotonic access structures do not allow for the usage of negative attributes (the usage of a NOT operator) and do not support the hierarchical relation of attributes. All these limitations hinder the expressiveness of ABE policies. Furthermore, attribute revocation and policy change management are still issues with ABE. All of these issues create a significant barrier to implementing ABE as part of industrial data sharing schemes, as they necessitate significant investment during the system’s development and policy management.

Access control models such as RBAC and ABAC offer standardized expressive policy languages such as XACML or NGAC, as well as easier ways to efficiently revoke access to attributes and add new ones to existing policies. However, for outsourced storage applications, such as cloud environments, pure RBAC or ABAC schemes are not sufficient. These access control models rely on fully trusted domains where data are expected to pass through access control before reaching the final user. When the data are outsourced to the cloud service, the data owner loses control, and it is susceptible to curious providers or leaks.

5. RESULT

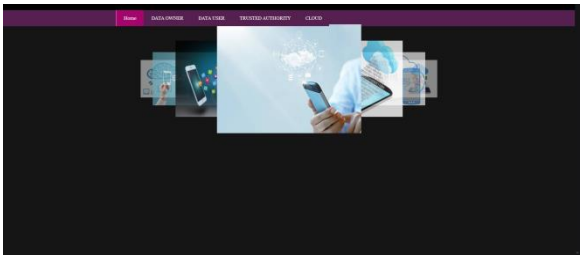

Fig 3. Home Page

Figure 3 Homepage of a project we have a four modules in our project.

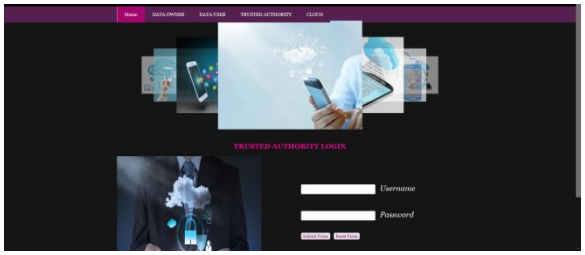

Fig 4. Trusted Authority Login Page

Figure 4 trusted authority of a login page has a shown in the figure.

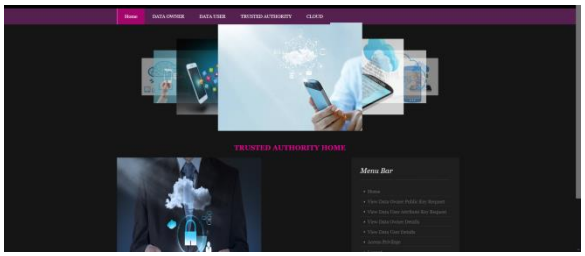

Fig 5. Trusted Authority Home Page

Figure 5 trusted authority of a home page has a shown in the figure.

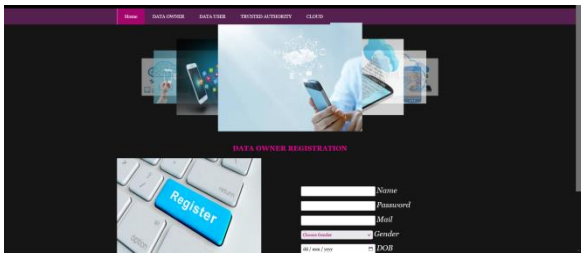

Fig 6. Data Owner Register Page

Figure 6data owner of a register page has a shown in the figure.

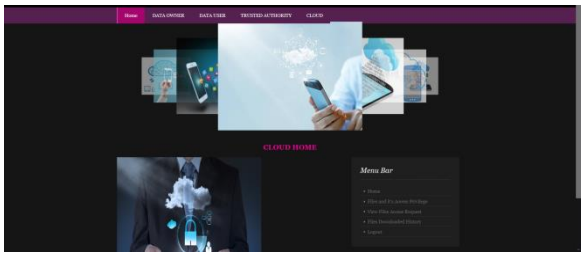

Fig 7:Cloud Home Page

Figure 7cloud homepage access has a shown in the figure.

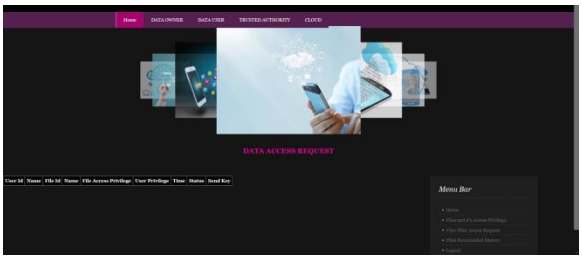

Fig 8. Cloud Server Data Access Request Page

Figure 8: Cloud server data has a request to send a data to a user.

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In this work, we examine industrial settings and the data sharing requirements they entail, and we identify the following essential characteristics of safe data sharing in those settings: 1) Access revocation; 2) expressiveness; 3) dynamic; 4) ease of management; and 5) simplicity of implementation. We introduce attribute-based cryptographic elements to assure data privacy in untrusted domains, and we motivate the usage of ABE-enabled ABAC, an attribute-based model that borrows concepts from ABAC models and architectures, to achieve such features. We carried out a cutting-edge investigation in the framework of the specified desirable attributes to find the weaknesses and pertinent mechanisms of each distinct technique, i.e., ABAC and ABE.

We discovered that effective mechanisms for enforcing expressive and fine-grained regulations over data hosted on private or public cloud servers are a major draw for ABAC research. Additionally, we compared SoTA strategies for data sharing with ABE and found shortcomings in the models' expressiveness and usability.

Additionally, in order to fill in the highlighted research gaps, we created a conceptual architecture that suggests how the pertinent processes should cooperate. This architecture makes an ABAC policy decision point a key component of the encryption process, allowing it to leverage its capabilities. When a request for data consumption or encryption is made, the PDP provides attribute and policy responses to the encryption and key generation method. This makes it possible for a central component to oversee and manage the ABE mechanisms while taking advantage of the adaptability and expressiveness of the standard policy languages used by ABAC. We conclude the study with a discussion of the deployment implications for the data owners and an examination of the knowledge gaps within the framework of our design. The primary area of uncertainty in this design pertains to the conversion

of ABAC access policies into practical parameters within the ABE encryption and decryption process.

6.2 FUTURE ENHANCEMENT

The interface between the PDP and the encryption and key generation algorithms has to be formally defined and implemented for future work. In this interaction, the decryption keys of the data are generated according to its assigned attributes; correspondingly, the decryption keys of the users are created according to their identities and related qualities. In addition, a study should be conducted to ascertain the consequences of applying the KP-ABE or CP-ABE methods for this architecture. Apart from the qualitative benefits in the context of industrial data sharing, there could be performance ramifications because essential generation processes run continuously when consumers request data. The conversion of access policies into access structures may happen more quickly or more slowly than the necessary attribute summarization for encryption and key generation, depending on the internal PDP process. Therefore, the optimal configuration should be ascertained through quantitative performance analysis.

REFERENCES

[1]. A. Beimel, "Secret-sharing schemes: A survey", Proc. Int. Conf. Coding Cryptol., pp. 11-46, 2011.
[2]. A. Elliott and S. Knight, "Role explosion: Acknowledging the problem" in Software Engineering Research and Practice, Princeton, NJ, USA: Citeseer, pp. 349-355, 2010.
[3]. A. Sahai and B. Waters, "Fuzzy identity-based encryption" in Advances in Cryptology–EUROCRYPT, Berlin, Germany: Springer, pp. 457-473, 2005.
[4]. B. Lang, J. Wang and Y. Liu, "Achieving flexible and self-contained data protection in cloud computing", IEEE Access, vol. 5, pp. 1510-1523, 2017.
[5]. B. Lang, R. Xu and Y. Duan, "Self-contained data protection scheme based on CP-ABE", Proc. Int. Conf. E-Business Telecommun., pp. 306-321, 2013.
[6]. C. Alliance, "Security guidance for critical areas of focus in cloud computing V3. 0", Cloud Secur. Alliance, vol. 15, no. 15, pp. 1-176, Nov. 2011.
[7]. C. S. Jordan, Guide to Understanding Discretionary Access Control in Trusted Systems, Collingdale, PA, USA: DIANE, 1987.
[8]. C.-C. Lee, P.-S. Chung and M.-S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments", Int. J. Netw. Secur., vol. 15, no. 4, pp. 231-240, Jul. 2013.
[9]. D. F. Ferraiolo, L. Feldman and G. A. Witte, Exploring the next generation of access control methodologies, 2016.
[10]. D. Ferraiolo, R. Chandramouli, D. Kuhn and V. Hu, "Extensible access control markup language (XACML) and next generation access control (NGAC)", Proc. ACM Int. Workshop Attribute Based Access Control, pp. 13-24, Mar. 2016.
[11]. D. Huang, Q. Dong and Y. Zhu, Attribute-Based Encryption and Access Control, Boca Raton, FL, USA: CRC Press, 2020.
[12]. D. R. Kuhn, E. J. Coyne and T. R. Weil, "Adding attributes to role-based access control", IEEE Comput., vol. 43, no. 6, pp. 79-81, Jun. 2010.
[13]. D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control", ACM Comput. Surveys, vol. 49, no. 4, pp. 1-45, Dec. 2017.
[14]. D. Wee, R. Kelly, J. Cattel and M. Breunig, "Industry 4.0-how to navigate digitization of the manufacturing sector", McKinsey Company, vol. 58, no. 58, pp. 7-11, Apr. 2015.
[15]. G. S. Mahmood, D. J. Huang and B. A. Jaleel, "A secure cloud computing system by using encryption and access control model", J. Inf. Process. Syst., vol. 15, no. 3, pp. 538-549, 2019.
[16]. International Data Space—Reference Architecture Model, 2019, [online] Available: https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf.
[17]. J. Callas, L. Donnerhacke, H. Finney, D. Shaw and R. Thayer, OpenPGP message format, 2007.
[18]. J. Weise, "Public key infrastructure overview", Sun BluePrintsOnLine, vol. 108, pp. 1-27, Aug. 2001.
[19]. L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, et al., "Security and privacy for storage and computation in cloud computing", Inf. Sci., vol. 258, pp. 371-386, Feb. 2014.
[20]. L. Zhou, V. Varadharajan and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage", IEEE Trans. Inf. Forensics Security, vol. 8, no. 12, pp. 1947-1960, Oct. 2013.
[21]. M. Copeland, J. Soh, A. Puca, M. Manning and D. Gollob, Microsoft Azure, 2015.
[22]. M. Falkenthal, F. W. Baumann, G. Grünert, S. Hudert, F. Leymann and M. Zimmermann, "Requirements and enforcement points for policies in industrial data sharing scenarios", Proc. 11th Symp. Summer School Service-Oriented Comput., pp. 1-14, 2017.
[23]. M. Joshi, S. Mittal, K. P. Joshi and T. Finin, "Semantically rich oblivious access control using ABAC for secure cloud storage", Proc. IEEE Int.

Conf. Edge Comput. (EDGE), pp. 142-149, Jun. 2017.

[24]. P. A. Loscocco, S. D. Smalley, P. A. Muckelbauer, R. C. Taylor, S. J. Turner and J. F. Farrell, "The inevitability of failure: The flawed assumption of security in modern computing environments", Proc. 21st Nat. Inf. Syst. Secur. Conf., pp. 1-12, 1998.

[25]. Q. Huang, Y. Yang and M. Shen, "Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing", Future Gener. Comput. Syst., vol. 72, pp. 239-249, Jul. 2017.

[26]. R. Basnet, S. Mukherjee, V. M. Pagadala and I. Ray, "An efficient implementation of next generation access control for the mobile health cloud", Proc. 3rd Int. Conf. Fog Mobile Edge Comput. (FMEC), pp. 131-138, Apr. 2018.

[27]. R. Ostrovsky, A. Sahai and B. Waters, "Attribute-based encryption with non-monotonic access structures", Proc. 14th ACM Conf. Comput. Commun. Secur., pp. 195-203, Oct. 2007.

[28]. R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, "Role-based access control models", Computer, vol. 29, no. 2, pp. 38-47, 1996.

[29]. S. Godik and T. Moses, "OASIS extensible access control markup language (XACML)" in OASIS Committee Secification Cs-XACML-Specification-1.0, OASIS OPEN, Woburn, MA, USA, 2002.

[30]. S. Granneman, Amazon Web Services, St. Louis, St. Louis, MO, USA: AECCU Guide, Washington Univ, Dec. 2012.

[31]. S. Kamara and K. Lauter, "Cryptographic cloud storage", Proc. Int. Conf. Financial Cryptography Data Secur., pp. 136-149, 2010.

[32]. S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing", IEEE Trans. Inf. Forensics Security, vol. 11, no. 6, pp. 1265-1277, Jun. 2016.

[33]. T. Kanwal, A. A. Jabbar, A. Anjum, S. U. Malik, A. Khan, N. Ahmad, et al., "Privacy-aware relationship semantics–based XACML access control model for electronic health records in hybrid cloud", Int. J. Distrib. Sensor Netw., vol. 15, no. 6, pp. 1-24, 2019.

[34]. T. T. A. Dinh, W. Wenqiang and A. Datta, "City on the sky: Extending XACML for flexible secure data sharing on the cloud", J. Grid Comput., vol. 10, no. 1, pp. 151-172, Mar. 2012.

[35]. V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, et al., "Guide to attribute based access control (ABAC) definition and considerations (draft)", NIST Special Publication, vol. 800, no. 162, pp. 1-54, 2013.

[36]. V. C. Hu, D. R. Kuhn and D. F. Ferraiolo, "Attribute-based access control", Computer, vol. 48, no. 2, pp. 85-88, Feb. 2015.

[37]. What is Considered Personal Data Under the Eu GDPR?, Feb. 2019, [online] Available: https://gdpr.eu/eu-gdpr-personal-data/.

[38]. X. Xiao and Y. Tao, "Anatomy: Simple and effective privacy preservation", Proc. 32nd Int. Conf. Very Large Data Bases, pp. 139-150, 2006.

[39]. Y. Wang, L. Wei, X. Tong, X. Zhao and M. Li, "CP-ABE based access control for cloud storage", Information Technology and Intelligent Transportation Systems, pp. 463-472, 2017.

[40]. Y. Zhu, D. Huang, C.-J. Hu and X. Wang, "From RBAC to ABAC: Constructing flexible data access control for cloud storage services", IEEE Trans. Serv. Comput., vol. 8, no. 4, pp. 601-616, Jul./Aug. 2015.