International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Implementation of Cypher Text- Policy Attribute- Set-Based Encryption (CP-ASBE) in Cloud

Mr. Suresh S[1*], Dr. Rakesh Kumar Yadav[2]

Submitted: 15/05/2024 **Revised**: 27/06/2024 **Accepted**: 08/07/2024

Abstract: This research seeks to explore the feasibility of using Attribute-Based Encryption (ABE) techniques, especially Comparative Policy-Based Attribute-Based Encryption (CP-ABE) and Comparative Policy-Attribute-Based Security Environment (CP-ASBE) for accurate access control in cloud computing. The comparison of CP-ABE and CP-ASBE is made based on the following aspects: access policy flexibility, scalability, efficiency, expressiveness, security, revocation mechanisms, and real-world uses. The proposed CP-ASBE architecture is based on dynamic access control at the attribute level. It employs up-to-date tools such as OpenSSL, Perceptome, AWS, Azure, Python, Java, Jenkins, and the ELK Stack. This makes the system scalable, efficient, and cryptographically compliant, which is a solution to cloud security problems. Some assessment methods are system testing, risk assessment, and continuous assessment to ensure the system works effectively and securely. Possible future research directions are the enhancement of homomorphic encryption, blockchain, AI security, and post-quantum cryptography. These developments aim to improve cloud security's capacity to address new threats and the needs of various regulations, which in turn contributes to the advancement of data protection and privacy in the cloud.

Keywords: *Revocation Mechanisms, Dynamic Access Control, Cryptographic Standards, Modern Tools, Java, Attribute-Based Encryption (ABE)*

1. Introduction

Cipher Text-Policy Attribute-Set-Based Encryption (CP-ASBE) is a state-of-the-art cryptographic technique specially developed to improve the security of data in cloud computing. It enables the data owner to decide on the access policy of the encrypted data. This method, development of Cipher Text-Policy Attribute-Based Encryption (CP-ABE), has several characteristics. Such as hierarchical user structures and flexible attribute management, making it a perfect fit for the dynamic cloud environment. In CP-ASBE, the encryption process includes creating an access policy defining which attribute sets can decrypt the data. The data owner encrypts the information using a public key and the policy that has been provided. Only the users with the correct attribute sets can decipher the cyphertext's decryption process then verifies the user's attributes against the policy encoded in the ciphertext, granting access only if the attributes align with the policy (Wang et al., 2011). CP-ASBE offers a multitude of benefits, including a high level of data security and a scalable hierarchical attribute management system(Zhou et al., 2011).

[1]*Ph.D.Research Scholar, Department of Computer Science, Maharishi School of Engg. & Tech., MUIT University, Lucknow, U.P. E-Mail: sureshsalendra@gmail.com,*
[2]*Associate Professor, Department of Computer Science, Maharishi School of Engg. & Tech., MUIT University, Lucknow, U.P.*
***Corresponding Author:** Mr. Suresh S*
E-Mail: sureshsalendra@gmail.com

In healthcare, CP-ASBE can safeguard patients' information and limit access to qualified healthcare workers.

In government, it can secure sensitive information, ensuring that only authorized personnel can view it. In educational institutions, CP-ASBE can control access to academic materials, research data, and students' records based on the roles of students, teachers, and managers. CP-ASBE is a robust encryption scheme that enhances cloud data security through its high access control, scalability, flexibility, and efficiency (Yu et al., 2010).

Cloud computing is a new model of computing that offers IT-enabled capabilities, such as internet services, to external customers, with the characteristics of scalability and elasticity. This model consolidates and standardizes computing, storage, and networking capabilities to be provisioned as needed, similar to how utilities such as electricity or water are provided (Mell & Grance, 2011). The history of cloud computing can be dated back to the 1960s when the idea of utility computing was envisaged by John McCarthy (McCarthy, 1961). However, it was not until the availability of high-speed internet and the development of virtualization technologies in the early 21st century that cloud computing became feasible for commercial use. Today, cloud computing is categorized into three primary service models: IaaS, which is the provision of computer infrastructure through the internet; PaaS, which is a service that provides customers with an environment in which to develop, manage, and run applications without having to manage the underlying

hardware; and SaaS which is the delivery of software applications through the internet on a subscription model (Buyya et al., 2009).Cloud computing has several features that set it apart from conventional computing paradigms. These are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service (Armbrust et al., 2010). These features enable users to allocate computing capacities on-demand, access services over the network, share resources to serve many consumers, quickly grow resources based on demand, and measure resource utilization (Vaquero et al., 2008) (Marston et al., 2011).

Cloud computing has limitations, including security and privacy, data localization, regulation, and lock-in (Subashini & Kavitha, 2011). To reduce these risks, organizations need to employ security measures such as encryption, access controls, and security audits (Popovic & Hocenski, 2010). Some modern trends in cloud computing include hybrid and multi-cloud environments, the emergence of edge computing, and the use of AI and ML in cloud services (Zhang et al., 2010). Future directions are expected to be influenced by quantum computing, which is expected to solve problems that classical computers cannot solve (Mohapi & Mnkandla, 2018). Cloud computing is an innovative technology offering elastic, versatile, and affordable solutions that enable innovation and create new opportunities for organizations and people (Hashem et al., 2015).

Importance of data security in cloud environments

Several defining characteristics of cloud computing set it apart from conventional computing paradigms. These are on-demand self-service, broad network access, resource pooling, fast elasticity, and measured service (Armbrust et al., 2010). These features enable users to allocate computing capacities on demand, access services through the network, share resources to serve many consumers, quickly increase or decrease resources based on demand, and measure resource utilization (Vaquero et al., 2008). The advantages of cloud computing include cost, flexibility, and scalability. This means that organizations can cut down on capital expenditure by adopting an operational expenditure model and scaling resources up or down depending on the need (Marston et al., 2011).

▪ **Encryption Methods**

Encryption is one of the most critical aspects of contemporary data protection, which involves converting the information into a form that is virtually only possible to decipher with proper access rights. Encryption methods can be broadly classified into two main categories: There are two types of encryptions, symmetric encryption, and asymmetric encryption, which have their processes and uses.

- **Symmetric Encryption**

Secret key encryption, also called symmetric encryption, employs the same key for encryption and decryption. This method is very effective and appropriate for encrypting large amounts of information. Notable symmetric encryption algorithms include: Notable symmetric encryption algorithms include:

- **Advanced Encryption Standard (AES):**

AES is considered to be the most secure method of data encryption, and it is used all over the world. It works on fixed block size of 128 bits and can process keys of 128, 192, and 256 bits. AES is widely applied in different fields because of its stability and speed; it is used in file encryption, network protection, and secure communication (Daemen & Rijmen, 2001).

- **Data Encryption Standard (DES):**

DES is one of the first symmetric encryption algorithms standardized for general use. It uses a 56-bit key and works on 64-bits of data simultaneously. Although DES has been used historically, it is now considered insecure because of the relatively small key size that can be easily cracked by a brute force attack (FIPS PUB 46-3, 1999).

- **Triple DES (3DES):**

Triple DES is an improvement of DES by applying the DES algorithm three times to each data block using two or three different keys. This approach is much more secure than the standard DES but less efficient than AES (Schneier, 1996).

- **Asymmetric Encryption**

Asymmetric encryption, also known as public-key encryption, uses a pair of keys: an encryption key available to the public and a decryption key only known to the owner. This method solves the main distribution issue associated with symmetric encryption and allows for secure communication over insecure channels. Prominent asymmetric encryption algorithms include: Prominent asymmetric encryption algorithms include:

- **Elliptic Curve Cryptography (ECC):**

ECC provides the same level of security as RSA but with much smaller keys, which makes it more effective. ECC is based on mathematical principles of elliptic curves that are defined over finite fields. Because of its effectiveness and security, it is suitable for mobile devices, IoT, and other limited settings. (Miller, 1985).

- **Diffie-Hellman Key Exchange:**

While not an encryption algorithm, the Diffie-Hellman key exchange protocol is vital for generating a secret key between two parties over an insecure channel. This shared

key can then be used for symmetric encryption (Diffie & Hellman, 1976).

- **Hybrid Encryption**

Hybrid encryption combines the features of both symmetric and asymmetric encryption. Asymmetric encryption is usually used to securely exchange a symmetric key, which is then used to encrypt the actual data. This approach uses symmetric encryption for data processing and asymmetric encryption for key exchange, which provides a good compromise between efficiency and security for communication (Menezes et al., 1996).

- **Quantum-Safe Encryption**

Quantum computing threatens current encryption methods, especially those based on asymmetric algorithms like RSA and ECC, since quantum algorithms like Shor's algorithm can break them. Therefore, there is a rising concern about quantum-safe or post-quantum encryption techniques resistant to quantum computers. These methods include lattice-based cryptography, hash-based cryptography, and code-based cryptography (Bernstein, Buchmann & Dahmen, 2009).

2. Objectives

The primary objectives of this research are:

- To propose a system architecture for CP-ASBE in cloud environments
- Modern tools and technologies will be used to implement the proposed architecture
- To assess the effectiveness and security of the system that has been put in place

3.Literature Review

- **Attribute-Based Encryption (ABE)**

Attribute-based encryption (ABE) is a type of public key encryption that allows the encrypted data to be accessed selectively. Unlike other encryption techniques, where data is encrypted for specific users, ABE allows data to be encrypted based on attributes or policies, thus making it easier to secure data.

Attributes:

In ABE, attributes are features or qualities related to users or data. These can be used to represent user roles, access rights, or any other information that may be deemed relevant. For example, attributes might include "Role: Doctor," Department: Cardiology," or "Clearance Level: High.

Access Policies:

Use policies define the conditions under which data can be made available for use. These policies are stated in terms of logical formulas over attributes. For instance, a policy might specify that only users with the attributes "Role: Some of the records are only accessible to "Doctor" and "Department: Cardiology."

Types of ABE:

Key-Policy ABE (KP-ABE): The ciphertext is associated with a set of attributes in KP-ABE, while the user's private key is connected to an access policy. A user can decrypt the ciphertext if the characteristics of the ciphertext correspond to the access policy of the user's private key (Bethencourt et al., 2007).

Setup and Key Generation:

The system starts with a setup phase that produces a master public key and a master secret key. The master secret key is used to derive private keys for users based on the attributes of the user. The master public key is used to encrypt data in accordance with the access policies that have been set.

Encryption and Decryption:

During encryption, data is encrypted with the master public key and this is accompanied by an access policy. The generated ciphertext can be decrypted only by the users whose attributes match the access policy associated with the ciphertext. Decryption is done with the help of the user's private key, which consists of the user's attributes.

- **Advantages**

Fine-Grained Access Control:

With ABE, it is possible to define fine-grained and dynamic access control policies. Data owners can explain the circumstances under which data can be accessed, thus increasing protection and adherence to organizational standards (Goyal et al., 2006).

Scalability:

ABE is highly scalable, given the number of users and attributes. Encryption is based on attributes, not the user, so it is easier to implement access control in large systems with many users and changing roles.

Reduced Key Management Overhead:

In traditional public-key encryption, every user has to deal with two keys, the public and the private ones. ABE decreases key management by linking the keys with attributes, thus making the management of access rights easier (Chase, 2007).

Decentralized Access Control:

ABE also supports distributed access control, where multiple authorities can control their attributes and policies without a central authority. This primarily benefits distributed systems and multi-tenant applications (Lewko & Waters, 2011).

- **Limitations**

Performance Overhead:

As a result, ABE schemes are generally more computationally intensive than conventional encryption techniques. The encryption and decryption operations are more complex, which can be a problem in terms of performance, particularly in low-power environments (Waters, 2011).

Key Management Complexity:

Although ABE decreases the number of keys required, it complicates handling attributes and policies. The generation and distribution of attribute-based private keys are sensitive and must be done efficiently (Yu et al., 2010).

Revocation Challenges:

Revoking access rights in ABE is more complex than in other systems. Since access is based on attributes, to revoke a user's attribute, one has to change the access policies and possibly re-encrypt the data, which is cumbersome and resource-consuming (Boldyreva et al., 2008).

Security Assumptions:

The security of ABE schemes is based on the hardness of some mathematical issues like the Bilinear Diffie-Hellman problem. The security of ABE could be affected by future improvements in cryptography or computational power (Boneh & Boyen, 2004).

- **Ciphertext-Policy Attribute-Based Encryption (CP-ABE): Evolution and Development**

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is one of the most significant advancements in cryptographic methods, particularly those offering fine-grained access control over encrypted data. It has been established that CP-ABE can be used to secure data in complex and dynamic settings such as cloud computing, healthcare, and finance. This paper has discussed CP-ABE in detail, including its background, significant advancements, applications, and possible future directions for research. ABE was introduced by Amit Sahai and Brent Waters in 2005. Their work was the first to construct the foundation of an encryption system where decryption is based on the attributes associated with the user's private key (Sahai & Waters, 2005). This early proposal was based on Key-Policy Attribute-Based Encryption (KP-ABE), in which the access policy is embedded in the user's private key. This approach offered a new way of managing access to encrypted information using descriptive characteristics, which opened the door to more complex encryption models. The roles are reversed in CP-ABE: the ciphertext contains the access policy, and the user's private key is associated with attributes (Bethencourt et al., 2007). Some of the issues encountered in early adoption include computational intensity and the size of the ciphertexts and keys. Some methods that improve efficiency include attribute-based essential delegation and efficient pairing-based cryptographic operations (Waters, 2011).

Besides the efficiency enhancement, the decentralized CP-ABE schemes have also been studied to overcome the drawbacks of the single trusted authority. In the traditional CP-ABE systems, a central authority always manages attributes and keys, which can be a bottleneck and a single point of failure. This decentralization is especially useful in the distributed setting, for example, in multi-tenant cloud environments where access to data must be regulated (Lewko & Waters, 2011). Newer developments have provided practical solutions for attribute revocation, reducing the cost of changing access control policies and re-encrypting the data (Yu et al., 2010). These mechanisms improve the feasibility of CP-ABE in environments where users' roles and access rights are often updated. The revocation techniques are essential for the security and practicality of CP-ABE in real life (Yang et al., 2013).

The practical uses of CP-ABE are diverse across different fields because of the flexibility of the access control policies that can be implemented. In cloud computing, CP-ABE allows data owners to decide who can access their data through attributes (Yu et al., 2010). This is especially useful in multi-tenant cloud environments where data access must be well-regulated to meet security and compliance requirements. CP-ABE is a good solution for protecting cloud data since it is scalable and flexible. In the healthcare industry, CP-ABE is used to protect EHRs, which means that only the healthcare workers with the right attributes can access the patient's data. This preserves the patient's confidentiality while at the same time allowing the authorized personnel to share information easily (Li et al., 2013). The capability to implement detailed access control policies is essential in healthcare since the information is sensitive and should not be accessed by unauthorized personnel. In contrast, the right personnel should be allowed to access the information they need to treat patients. Financial institutions also employ CP-ABE to safeguard their financial data, which is only accessible to personnel with the correct roles and clearances. This assists in meeting the legal and regulatory standards and protecting sensitive financial data (Zhou et al., 2015). The fine-grained access

control that CP-ABE offers is crucial for financial institutions to control access to the data and meet regulatory standards. In the context of the Internet of Things (IoT), CP-ABE is used to control access to devices and data streams. CP-ABE can use attributes like device type, location, and operational status to provide secure and contextually appropriate access to IoT data (Ruj, Nayak, & Stojmenovic, 2011). The development of CP-ABE is continuous due to new security threats and the need to expand its usage in various scenarios. One of the most important future research areas is post-quantum security. Since quantum computing is a threat to current cryptographic schemes, there is a shift towards constructing secure CP-ABE schemes against quantum attacks. Post-quantum cryptographic methods have been developed to enhance CP-ABE security in the post-quantum world (Boneh et al., 2009).

Critical research area is the improvement of the privacy of attribute information and access policies. Some approaches include anonymous attribute-based encryption and policy hiding to enhance users' privacy and attributes while simultaneously providing high levels of access control (Liu et al., 2018) Current research aims to decrease the computational cost, decrease the size of the ciphertexts and keys, and design efficient revocation schemes for large-scale applications (Emura et al., 2009).

- **Comparative Analysis of CP-ABE and CP-ASBE**

CP-ABE and CP-ASBE are two complicated cryptographic methods designed to control access to the encrypted data at the micro level. While they are somewhat similar, they are also different, making them suitable for other purposes. The following table compares CP-ABE and CP-ASBE concerning aspects such as expressiveness of the access policy, scalability, efficiency, flexibility, and security.

Feature	CP-ABE	CP-ASBE	References
Access Policy Expressiveness	Supports expressive access policies using logical operators such as AND, OR, and threshold gates (Bethencourt, Sahai, & Waters, 2007).	Extends CP-ABE by allowing hierarchical access structures, enabling more granular and complex policy definitions (Wan, Liu, & Deng, 2012).	Bethencourt, Sahai, & Waters (2007); Wan, Liu, & Deng (2012)
Scalability	Limited by the complexity of the access policy and the number of attributes involved.	More scalable due to its hierarchical structure, which reduces the complexity of access policy management (Wan, Liu, & Deng, 2012).	Waters (2011); Wan, Liu, & Deng (2012)
Efficiency	Encryption and decryption operations can become computationally intensive as the number of attributes increases (Bethencourt, Sahai, & Waters, 2007).	Improved efficiency in attribute management and policy enforcement due to the hierarchical nature of attribute sets (Wan, Liu, & Deng, 2012).	Waters (2011); Wan, Liu, & Deng (2012)
Flexibility	Allows for flexible access control but can be cumbersome to manage in large-scale systems with many attributes (Yu et al., 2010).	Offers greater flexibility in managing attributes and access policies in large-scale and hierarchical environments (Wan, Liu, & Deng, 2012).	Yu et al. (2010); Wan, Liu, & Deng (2012)
Revocation Mechanisms	Traditional revocation mechanisms require re-encrypting data or updating keys, which can be inefficient (Yu et al., 2010).	More efficient attribute revocation mechanisms that minimize the overhead associated with updating policies and re-encrypting data (Wan, Liu, & Deng, 2012).	Yu et al. (2010); Wan, Liu, & Deng (2012)
Security	Provides robust security but may be vulnerable to certain attacks if not properly implemented (Bethencourt, Sahai, & Waters, 2007).	Enhanced security features due to the hierarchical organization of attributes, providing better resistance to certain types of attacks (Wan, Liu, & Deng, 2012).	Waters (2011); Wan, Liu, & Deng (2012)
Policy Update and Dynamic Attributes	Supports policy updates and dynamic attributes but with	More efficient handling of policy updates and dynamic attributes,	Yang et al. (2013); Wan, Liu, & Deng (2012)

	significant computational overhead (Yang et al., 2013).	reducing computational overhead (Wan, Liu, & Deng, 2012).	
Practical Applications	Widely used in cloud computing, healthcare, and financial services for fine-grained access control (Yu et al., 2010; Li et al., 2013).	Suitable for complex and hierarchical environments, such as large organizations and IoT ecosystems (Wan, Liu, & Deng, 2012).	Yu et al. (2010); Li et al. (2013); Wan, Liu, & Deng (2012)

Methodology

1. Understanding CP-ASBE Requirements:

CP-ASBE allows access control at the attribute level rather than the identity level. This implies that the architecture must support dynamic policies and multiple user attributes in cloud environments.

System Architecture Components:

- **Client Interfaces:** Interfaces for the user and the authentication of the user.
- **Critical Management Services:** Ensure the services of essential generation, distribution, and revocation are secure.
- **Encryption/Decryption Modules**: Modules for encrypting and decrypting the data based on attribute-based policies.
- **Integration with Cloud Storage:** Ensure secure contact points with cloud storage services for storing and retrieving encrypted data.

Data Flow and Security:

- Explain how data and communication flow to ensure data security and access rights in different cloud environments.
- Employ secure communication protocols such as TLS/SSL and potent forms of authorization and authentication to enhance data security and integrity.

Scalability and Performance:

- Ensure that the architecture is horizontally and vertically scalable to accommodate more users and data in the future.
- Ensure that resources are well utilized and implement measures to avoid overloading the system to optimize efficiency.

Security Measures:

- Adhere to the cryptographic best practices and policies for data encryption and key management.
- Implement strict access control measures that comply with the organizational security policies and the law.

Integration with Existing Cloud Services:

- Adherence to the APIs of the cloud provider and compatibility with other cloud applications.
- Ensure that data can be migrated between cloud services and that the services are interoperable.

Documentation and Communication:

- Suggest the architecture and make sure that diagrams and configuration details support it.
- It should contain recommendations for implementing the CP-ASBE system and the steps to configure and operate it in cloud environments.

2. Modern tools and technologies will be used to implement the proposed architecture

The deployment of the proposed architecture for CP-ASBE in cloud environments uses the current tools and technologies known for their efficiency, security, and scalability. The encryption and decryption of data using the proposed CP-ASBE algorithms are managed by OpenSSL and PyCryptodome (Raghav & Harit, 2020). AWS and Azure, with their reliable infrastructure services, offer a scalable platform that can support the integration and expansion of the CP-ASBE system (Amazon et al., 2021; Microsoft Azure, 2021). Python and Java, selected for their cloud deployment suitability, enable the creation of safe encryption modules and critical management services, ensuring the system's scalability (Python Software Foundation, 2021; Oracle, 2021). Updates are released as soon as possible through continuous integration and delivery with Jenkins or GitLab CI (Jenkins, 2021; GitLab, 2021). Prometheus and ELK Stack are used for monitoring and logging system performance, analysing deviations, and implementing security policies (Prometheus, 2021; Elastic, 2021).

- **Elaboration on Assessing System Effectiveness and Security**

Assessing the effectiveness and security of the implemented CP-ASBE system in cloud environments is critical to validate its operational integrity and resilience against potential threats. Effectiveness evaluation involves rigorous testing methodologies to measure the system's ability to enforce access control policies accurately and efficiently (Choi et al., 2018), ensuring that authorized users can access data while unauthorized attempts are appropriately denied. Security assessment

encompasses comprehensive vulnerability analysis, penetration testing, and compliance checks against established cryptographic standards (NIST, 2021) and regulatory requirements (EU GDPR, 2016). Continuous monitoring and auditing mechanisms are employed to detect anomalies, mitigate risks promptly, and maintain data confidentiality, integrity, and availability across distributed cloud infrastructures (Shamir et al., 2019). By systematically evaluating both effectiveness and security, this research aims to validate the robustness of the CP-ASBE implementation in safeguarding sensitive information and upholding organizational security objectives.

Challenges and future directions

Sophisticated solutions in cloud security include new technologies and approaches aimed at improving data security, confidentiality, and resistance to threats in the cloud. These are homomorphic encryption for computation on encrypted data, blockchain for data integrity, and AI for threat detection and response. Future research areas in cloud security include post-quantum cryptography to counter quantum computing attacks, edge computing security, privacy-preserving data analytics, IoT security, and global data protection regulation compliance. By exploring these applications and future directions, this research will help advance the field of cloud security and respond to new threats and opportunities in data protection and privacy.

Conclusion

The proposed CP-ASBE architecture for cloud environments is a comprehensive and robust framework that can enhance data security through attribute-based access control. The system is intended to be scalable and effective with the use of contemporary tools and technologies such as OpenSSL, PyCryptodome, AWS, Azure, Python, Java, Jenkins, and the ELK Stack, as well as to conform to the standards and legislation of cryptography. The systematic approach to assessing the system's performance and security through testing, identifying vulnerabilities, and monitoring speaks of its readiness to counter potential threats. Moreover, identifying future research areas like homomorphic encryption, blockchain, and AI-based security solutions, along with the development of post-quantum cryptography and edge computing, helps the CP-ASBE system address emerging threats and enhance cloud security to safeguard data and support organizational security objectives.

References

[1] Wang, C., Wang, Q., Ren, K., Cao, N., & Lou, W. (2011). Toward Secure and Dependable Storage Services in Cloud Computing. IEEE Transactions on Services Computing, 5(2), 220-232.

[2] Zhou, Z., Huang, D., & Wang, Z. (2011). Efficient Privacy-Preserving Cipher Text-Policy Attribute-Based Encryption in Cloud Computing. Proceedings of the 2011 IEEE International Conference on Computer and Information Technology, 17-25.

[3] Yu, S., Ren, K., Lou, W., & Li, J. (2010). Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. Proceedings of the IEEE INFOCOM 2010 Conference on Computer Communications, 1-9.

[4] Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. National Institute of Standards and Technology, Special Publication 800-145.

[5] McCarthy, J. (1961). Speech at the MIT Centennial. Massachusetts Institute of Technology.

[6] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599-616.

[7] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58.

[8] Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review, 39(1), 50-55.

[9] Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—The business perspective. Decision Support Systems, 51(1), 176-189.

[10] Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 34(1), 1-11.

[11] Popovic, K., & Hocenski, Z. (2010). Cloud computing security issues and challenges. Proceedings of the 33rd International Convention MIPRO, 344-349.

[12] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1), 7-18.

[13] Mohapi, L., & Mnkandla, E. (2018). Quantum computing: a review of the state of the art. Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD).

[14] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47, 98-115.

[15] Daemen, J., & Rijmen, V. (2001). The design of Rijndael: AES-the advanced encryption standard. Springer-Verlag.
[16] FIPS PUB 46-3. (1999). Data Encryption Standard (DES). National Institute of Standards and Technology.
[17] Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons.
[18] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.
[19] Miller, V. S. (1985). Use of elliptic curves in cryptography. Advances in Cryptology—CRYPTO '85, 417, 417-426.
[20] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.
[21] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.
[22] Bernstein, D. J., Buchmann, J., & Dahmen, E. (2009). Post-quantum cryptography. Springer-Verlag.
[23] Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-policy attribute-based encryption. 2007 IEEE Symposium on Security and Privacy (SP '07), 321-334.
[24] Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06), 89-98.
[25] Chase, M. (2007). Multi-authority attribute-based encryption. Proceedings of the 4th Theory of Cryptography Conference (TCC '07), 515-534.
[26] Lewko, A., & Waters, B. (2011). Decentralizing attribute-based encryption. Advances in Cryptology – EUROCRYPT 2011, 568-588.
[27] Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Public Key Cryptography – PKC 2011, 53-70.
[28] Boldyreva, A., Goyal, V., & Kumar, V. (2008). Identity-based encryption with efficient revocation. Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS '08), 417-426.
[29] Boneh, D., & Boyen, X. (2004). Efficient selective-ID secure identity-based encryption without random oracles. Advances in Cryptology – EUROCRYPT 2004, 223-238.
[30] Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. Advances in Cryptology – EUROCRYPT 2005, 457-473.
[31] Li, M., Yu, S., Zheng, Y., Ren, K., & Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Transactions on Parallel and Distributed Systems, 24(1), 131-143.
[32] Yang, K., Jia, X., Ren, K., Zhang, B., & Xie, R. (2013). DAC-MACS: Effective data access control for multiauthority cloud storage systems. IEEE Transactions on Information Forensics and Security, 8(11), 1790-1801.
[33] Wan, Z., Liu, J., & Deng, R. H. (2012). HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing. IEEE Transactions on Information Forensics and Security, 7(2), 743-754.
[34] Boneh, D., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2009). Fully homomorphic encryption without bootstrapping. Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 169-178.
[35] Liu, J. K., Au, M. H., Huang, X., & Susilo, W. (2018). Practical privacy-preserving access control over encrypted data in cloud computing with limited key leakage. IEEE Transactions on Information Forensics and Security, 10(8), 1590-1601.
[36] Amazon Web Services. (2021). AWS. Retrieved from https://aws.amazon.com/
[37] Elastic. (2021). Elastic Stack: Elasticsearch, Kibana, Beats, and Logstash. Retrieved from https://www.elastic.co/
[38] GitLab. (2021). GitLab CI. Retrieved from https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/
[39] Jenkins. (2021). Jenkins. Retrieved from https://www.jenkins.io/
[40] Microsoft Azure. (2021). Azure. Retrieved from https://azure.microsoft.com/
[41] Oracle. (2021). Java. Retrieved from https://www.oracle.com/java/
[42] Prometheus. (2021). Prometheus. Retrieved from https://prometheus.io/
[43] Python Software Foundation. (2021). Python. Retrieved from https://www.python.org/
[44] Raghav, H., & Harit, A. (2020). PyCryptodome: A Python cryptographic library. Journal of Open Source Software, 5(47), 1956. https://doi.org/10.21105/joss.01956
[45] Choi, J., Park, J., & Lee, H. (2018). Effective access control scheme using attribute-based encryption in cloud computing. Journal of Supercomputing, 74(8), 3493-3508. https://doi.org/10.1007/s11227-018-2437-1
[46] EU GDPR. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free

movement of such data. Official Journal of the European Union. Retrieved from https://eur-lex.europa.eu/eli/reg/2016/679/oj

[47] NIST. (2021). NIST Special Publication 800-175B: Guideline for using attribute-based access control (ABAC) in information sharing environments (ISE). National Institute of Standards and Technology. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-175B/final