Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 21 No. 8 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v21i08.55031

PAPER

Enhance the Security of Data Based on Server-Side Encryption in a Cloud Environment

Waleed Almuseelem(✉)

Faculty of Computing and Information Technology (FCIT), University of Tabuk, Tabuk, Saudi Arabia

waleedalmuseelem@ut.edu.sa

ABSTRACT

Cloud computing has gained popularity due to its advantages for users and providers. This study examined and analyzed the effectiveness of server-side encryption (SSE) aimed at enhancing the security of cloud data storage and transfer. The researcher used Amazon Web Services key management service keys (AWS KMS) to perform the simulation over the AWS platform. The encryption method was integrated with identity and access management (IAM) to help with accessibility control. The findings demonstrated that server-side encryption (SSE), when merged with KMS and IAM roles, helps with restricted access and secures data from unauthorized users. The results show that unauthorized users only view ciphertext rather than plain text, highlighting the security feature of server-side encryption.

KEYWORDS

cloud security, data security, server-side encryption (SSE), data storage, access control, authentication

1 INTRODUCTION

Cloud computing refers to a model system that enables the on-demand delivery of computing resources such as database applications, computing power, servers, storage, and networks [1]. The Internet delivers these computing resources via a cloud services platform with pay-based usage. The cloud is a computer located somewhere a user can access via the Internet. The cloud consists of server computers located in larger data centers in different locations globally.

Cloud computing's popularity is based on the benefits that users and cloud providers both enjoy. Technology has helped reduce complexity regarding relying on traditional infrastructure, such as physical databases and other non-virtualized infrastructures. The migration of services to the cloud has helped organizations reduce energy consumption that could have powered non-virtualized infrastructures such as physical databases. Reliance on servers provided by third parties helps minimize operation costs. The reduced fees are passed to customers who receive

Almuseelem, W. (2025). Enhance the Security of Data Based on Server-Side Encryption in a Cloud Environment. *International Journal of Online and Biomedical Engineering (iJOE)*, 21(8), pp. 75–103. https://doi.org/10.3991/ijoe.v21i08.55031

Article submitted 2025-02-28. Revision uploaded 2025-05-05. Final acceptance 2025-05-05.

services at a reduced cost [1]. Recently, the charging model offered by cloud providers based on usage has increased the popularity of cloud computing rather than reliance on a dedicated physical infrastructure.

One significant benefit of migrating services to the cloud is reducing system complexity and helping users trade fixed expenses for variable expenses [2]. A fixed expense entails the company's funds to acquire and maintain physical assets. In contrast, a variable expense refers to an expense that the person who bears the cost can avoid. Cloud technology has removed the need for organizations to invest in data centers to access computing resources. Instead, users need to pay only for the computing resources utilized based on how much they use. This helps users to save money. Cloud computing allows public cloud providers, including AWS, to attain economies of scale based on the many users of their platform from around the globe. These massive economies enable providers to offer lower pay-as-you-go prices. Cloud technology allows users to scale the accessed cloud computing services quickly, helping avoid expenses of expensive idle resources or reduce problems due to insufficient capacity of the resources. Cloud computing also helps organizations increase speed and agility regarding access to computing resources, as they are a click away. Cloud computing also allows users to deploy their applications globally in minutes. The availability of multiple cloud provider regions helps lower latency, improving customer experience worldwide.

The three main cloud computing deployment methods are private, public, and hybrid. Computing resources associated with a private cloud are only accessible to a particular authorized group. A public cloud allows users to share computing resources regardless of their location. A hybrid cloud includes computing resources only accessible to a specific group of users and resources that can be shared with the general public.

Cloud computing services deployment can be categorized into IaaS, PaaS, and SaaS [3]. IaaS allows users to manage their servers, which can be virtual or physical, as well as the OS (Linux or Microsoft Windows). With PaaS, the public cloud service provider manages the underlying infrastructure (hardware and OS), enabling customers to use services. Additionally, PaaS offers a framework that allows developers to create customized applications efficiently. SaaS allows users to manage their files while the service provider manages all data centers, storage, servers, patching, and maintenance. With SaaS, users can only handle the software. Dropbox and Facebook are some examples of SaaS.

The SaaS field has led to Database as a Service (DBaaS), which helps provide better functions regarding database management [4]. DaaS's popularity rise is based on the change from the traditional data management client-server architecture to a web-based architecture where users of cloud platforms migrate their data to public service providers who provide underlying infrastructure, including the security of data centers, to help secure the data. Migrating databases to the cloud has several benefits, including cost reduction, better services, and remote online access. However, adopting DaaS presents security as the main critical issue, as data can be compromised. Cloud providers have adopted encryption methods to help prevent loss of confidentiality in cloud databases.

Cloud providers allow the usage of cryptography methods such as encryption to uphold the integrity of customers' data. Cryptography is the art used to create encrypted information and secure data exchange. Encryption is considered a significant security solution due to various benefits. Data confidentiality is one key reason, as cloud data in the cloud is inaccessible to unauthorized users [1]. Suppose there is an event of unauthorized access; encryption provides another layer of protection as the data will need decryption to make sense of the information. Data encryption also helps users trust cloud providers since they can control their data if there is a data breach. Encryption

can also facilitate data sharing and collaboration in the cloud by ensuring authorized handlers possess the decryption key and can view the data as plain text.

Cloud storage providers offer various encryption methods, including client-side encryption, where users use their key for data encryption, and server-side encryption, where users upload the data and the cloud provider encrypts the data. In client-side encryption, data is encrypted before it is sent to the server. The user has full control over the encryption keys. Data remains safe; suppose the cloud platform used to store the data is breached, and there is unauthorized access to the data. In server-side encryption, the users hold their key, but the server encrypts the data [4]. Users may not have full control over encryption keys, as the service provider typically manages the encryption keys. Additionally, there is end-to-end encryption, where the sender performs encryption and decryption, and the receiver performs these via public/private keys. Server-side encryption entails using a secure key to encrypt the data via encryption methods at the server, but the customers hold the encryption key. Server-side encryption entails the conversion of plaintext to cyphertext in the server while data is at rest.

Cloud providers are experiencing an increased demand for cloud data storage and transfer. The increase in demand poses a challenge for cloud providers in providing a secure, reliable, trusted cloud platform solution. Some of the most popular public cloud providers that offer SaaS include Amazon Web Service (AWS), GCP, and Microsoft Azure. This paper focused on AWS SSE using the KMS to create encryption keys. According to [5], Amazon Web Services key management service keys (AWS KMS) is a service that allows customers to create, edit, and delete encryption keys. AWS KMS operates as a cryptographic service provider that protects data. To protect data keys, AWS KMS utilizes authenticated encryption. The AWS server-side encryption stores the encryption key in a storage unit that is different from the one that stores the data. Amazon S3, when integrated with AWS KMS, provides server-side encryption (SSE) via the KMS keys (SSE-KMS).

The paper aims to evaluate the effectiveness of SSE in enhancing the security of cloud data storage and transfer. The researcher will use Amazon S3 SSE-KMS to perform the simulation. The results may help provide various appropriate security measures for cloud data storage and management.

2 LITERATURE REVIEW

2.1 Challenges to cloud computing

According to [11], there is a growing demand for data storage and management due to the rise in data among users and businesses. The swift technological advancement has enabled cloud computing to be an ideal solution. However, cloud service providers face the challenge of providing a secure, reliable, trusted, and user-friendly data storage and management solution. [9] argues that cloud computing relies on the Internet, implying privacy and data protection must be prioritized. Data exposure or loss may negatively impact an organization's reputation and confidence. According to [9], data leakage prevention presents 88% of the significant challenges. Additionally, privacy and data remoteness impact security problems by 92%. Data transfer to the cloud may be subject to vulnerability, such as unlawful alteration, and there is a need to ensure data validity is continuously protected. Data availability at any time of request presents another challenge to cloud providers in retaining access to cloud resources. Cloud providers may be subjected to vendor lock-in conditions where they fail to satisfy customers' needs regardless of upgrades or

improvements of services or infrastructure. The data stored in the cloud may be vulnerable to security threats. Various solutions can be implemented to safeguard this data, including encryption, certification, and intrusion detection systems, which are poorly implemented. Integrating with a cloud provider's two or more processes is also challenging to ensure data is shared and adequately used for smooth operation.

2.2 Security issues in cloud computing

Many reviewed literatures identified data security as the central issue of cloud computing. According to [9], data security is the biggest issue regarding providing SaaS. Cloud providers keep their users' files, and the main concern is security leaks, which might subject data to breaches or cyberattacks. According to [1], security as the primary concern of cloud challenges may incur vulnerabilities based on data exchange security, secure interfaces based on data transfer, storage, and recovery, policies to ensure data privacy, and access control based on compliance auditors and security managers. [10] categorized cloud security vulnerabilities to include safe data migration, safe cloud data storage, and user control via permissions, which restrict the accessibility of resources and services associated with cloud data storage and transfer. [4] argued that using SaaS requires cloud service providers to prioritize data security. The following are some of the cyber-threat challenges to securing data.

Cyberattacks in DBaaS. Cyber threats in DBaaS involve the use of unauthorized access to find entry into the cloud databases and the use of private information for malicious activities [15]. Hackers can gain access to private data and destroy or publish it to compromise organizational or personal data. Since databases store a set of data through the network, access to information or data kept in databases is more manageable [15]. As such, cyber threats such as SQL injections can quickly occur where intruders or hackers can inject unauthorized applications into the database systems. Data sharing across database systems facilitates security threats, especially when their database users violate security policies. In cloud databases, hackers or intruders can quickly gain unauthorized access to cloud data via the Internet. These security challenges affect confidentiality, privacy, interpretation, and data availability in the systems.

Database as a Service allows replication of shared data in distributed database systems, presenting security challenges. The replicated data in the distributed databases makes it easier for insiders or intruders to copy, share, and synchronize the available data between the databases for malicious activities. This unauthorized access is associated with a lack of security features such as authorization and authentication, which facilitates cyber threats such as cross-site scripting and SQL injections [15].

A firewall is a software-based network security device that restricts unauthorized access to a network [15]. It uses predefined rules to allow or deny incoming or outgoing traffic. One common technique hacker's use to bypass firewalls is encrypted injection attacks, often initiated via phishing emails. Additionally, SQL injection attacks are a prevalent method for hackers to circumvent a user's database or system firewall by injecting malicious SQL code. SQL injection is one of the most widely used techniques in web hacking.

Cloud computing platforms are also vulnerable to phishing attacks [15]. Phishing occurs when users are deceived into sharing sensitive information, such as login details, via fraudulent links or attachments sent via email. Actors of phishing attacks create malicious messages with links that may appear legitimate, enticing the recipient to use the link, which exposes sensitive information. The links in the phishing emails are commonly embedded with malware that redirects users to fake websites that may appear legitimate. The users who use these fake websites end up exposing

their login information. The tactics employed by phishers continue to evolve to bypass established security measures, making this a significant issue. Attackers are always looking for new vulnerabilities and ways to conduct their attacks. Phishers may use these methods to make authorized cloud users expose their sensitive data, such as encryption keys and login information. An example highlighting the phishing threat occurred on July 25, 2017, when Florida Health Kids Corporation staff members received phishing emails [16]. Those who responded to these emails inadvertently granted the attackers access to their sensitive information.

Another method hackers employ is social engineering, where they pose as tech assistants or customer service representatives to gain a user's trust. By exploiting this trust, the hacker can perform malicious activities. Hackers bypass firewalls to access sensitive data, systems, and personal information such as bank details. Hackers may steal or destroy information and systems once they gain unauthorized access by circumventing firewalls [15].

Solution to data security challenges. The reviewed literature highlighted the various solutions to data security challenges, including intrusion detection systems (IDS), conducting penetration tests, and encryption. Most studies reviewed indicated encryption as the key solution to secure cloud data stored in online databases or services such as Amazon S3. For instance, [1] reviewed the various cryptography algorithms to help address security concerns as the central issue in cloud storage. [3] Assume encryption is critical for cloud computing since most cloud operations are based on data transfer via the internet during data migration to cloud servers or between the various cloud resources or services. According to [9], data encryption is best to ensure privacy and data confidentiality in cloud environments.

Encryption methods. Encryption entails converting data into another form where users are the only ones able to decrypt the converted data via respective keys or another access mechanism. According to [3], encryption can be categorized into public-key encryption, private-key encryption (asymmetric encryption), and symmetric encryption. Symmetric key encryption uses a single key for the encryption process. In contrast, asymmetric encryption involves the creation of two keys: an encryption public key and a decryption private key.

1. **Symmetric encryption:** According to [4], a symmetric type of encryption utilizes the same key to decrypt and encrypt cloud data, as highlighted in Figure 1 below. Symmetric encryption is generally used for encrypting databases and storage volumes since the overhead of computing in asymmetric encryption is higher than that of symmetric encryption. All public cloud service providers use the standard symmetric algorithm AES-256. On the other hand, symmetric encryption faces one vulnerability disadvantage based on using a single key. Unauthorized access to the encryption key exposes the cloud data.

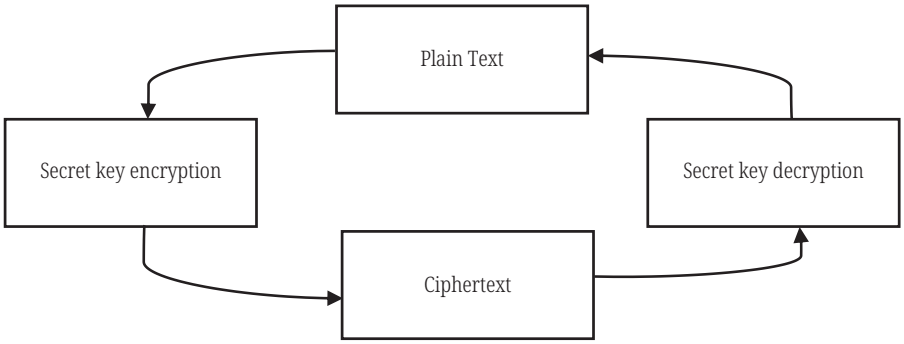

Fig. 1. Symmetric encryption via a single

[23] Explored the performance of symmetric encryption methods in terms of encryption speed and CPU utilization. The encryption speeds of 10 symmetrical algorithms (AES, SEED, DES, RC2, DESede, IDEA, RC4, BlowFish, RC6, and XTEA) were examined based on different file sizes ranging from 1MB to 1GB. The results demonstrated AES emerged among the best in terms of performance and security.

According to [24], symmetric encryption is faster than asymmetric encryption for a given throughput, supposing they are subjected to similar sizes in terms of computing resource allocation. This is based on the fact that symmetric encryption uses the same key, whereas asymmetric encryption relies on a pair of keys. [25] demonstrated that symmetric encryption was superior to asymmetric encryption in terms of encryption speed for AI-based networks.

2. **Asymmetric encryption:** With asymmetric encryption, a set of two key pairs is used. One key is known as a private key, and the other is a public key, as highlighted in Figure 2 below. Asymmetric encryption is used between a server and a client to ensure end-to-end encryption. For instance, a website provider is responsible for keeping the private key via a signed certificate from a CA, and the public key is distributed to clients when they visit the website [4]. According to [3], one main benefit of asymmetric encryption is that the encryption key distribution is better managed and in a more secure manner. Suppose an unauthorized user possesses the public key; the encrypted data remains confidential as the key cannot decrypt it. One good practice to ensure the keys remain secure is establishing an effective strategy to distribute the keys. According to [6], ECC, RSA, and Diffie-Hellman are the three most popular asymmetric algorithms.

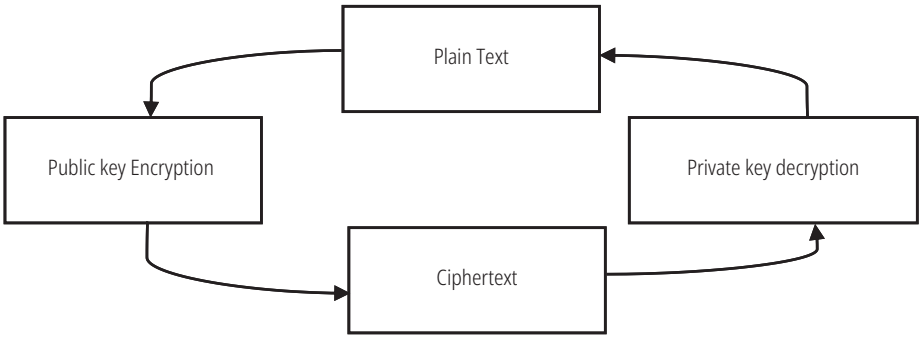

Fig. 2. Asymmetric encryption via public-private key pair

RSA: RSA was named after its inventors (Rivest, Shamir, and Adelman). According to [6], the algorithm is ideal for securing data in cloud-based environments. Data of users of the RSA algorithm is first encrypted, and then the ciphertext is stored in the cloud. In RSA, the public key is accessible to everyone, and the end user only knows the private key. The generation of these keys relies on a mathematical formula involving prime number multiplication. RSA cryptography uses a significant number that is challenging to separate into two big prime numbers to strengthen the code. Decryption consists of multiplying the cipher texts to identify the plain text.

The RSA algorithm key's long length makes encryption and decryption operations slower [26]. On the other hand, security depends on the key size of the RSA algorithm, making it hard to break the algorithm. This is why the RSA algorithm is used in online banking, as there is a need to encrypt sensitive information [6].

3. **Elliptic curve cryptography (ECC):** According to [7], ECC is a robust cryptographic strategy. ECC is similar to RSA, as both algorithms use mathematical computations to encrypt data. However, ECC key generation relies on the mathematics

of elliptic curves rather than the prime factorization used in RSA. Therefore, ECC creates keys that are more difficult to crack mathematically. The key generated in ECC involves a code system that consists of plain text, ciphertext, and a key. The key can be private, public, or partially private and partially public. ECC's popularity is rising based on the minor size of the key and its ability to maintain security. Proposed a hybrid method involving ECC that focused on an e-healthcare system. The results demonstrated the algorithm increased security. However, a high level of computation and communication overhead was a major issue.

Table 1 compares the security strength between RSA and ECC algorithms. Table data illustrates that ECC is better than RSA in terms of security strength. ECC is for environments with limited computing resources, such as IoT applications.

Table 1. The approximate key comparison table [27]

RSA Key Size(Bits)	ECC Key Size (Bits)	Key Size Ratio	Times to Break (MIPS Year)
512	106	1:5	10^4
768	132	1:6	10^8
1024	160	1:7	10^12
2048	210	1:10	10^20

4. **Diffie-Hellman:** This algorithm is based on swapping encryption keys. The swapping involves a shared hidden key published via swapping the shared keys between two end users [8]. A third key, known as the session key, is generated, making it hard for an unauthorized user to reproduce. The communicating parties share two public numbers. The shared numbers involve a sizeable prime integer and its primitive root a. The Diffie-Hellman algorithm makes it difficult for unauthorized users to calculate its distinct logarithms. However, Diffie–Hellman protocol fails to authenticate the communicating parties, exposing it to a man-in-the-middle attack. Solutions to the vulnerability entail using public certificates and digital signatures as authentication mechanisms [8]. Diffie-Hellman provides various solutions for cloud environments, including key components of transport layer security (TLS) and SSH protocol to enable secure remote access in VPNs and in VoIP applications such as Managed VoIP Phone Systems.

Hardware security modules. According to [19], a hardware security module (HSM) entails physical devices that add an extra layer of protection to cryptographic keys, including private-public key pairs used in encryption operations. The rise of cloud computing has led to an increasing demand for HSMs in the insurance and financial industries. These industries rely on key HSM functions, including key creation, key management, hashing, and data encryption and decryption. According to [20], hardware key versions are more expensive than software keys, but they provide enhanced security. Google Cloud provides HSM services that help customers perform cryptographic operations such as key generation [20]. Google's HSM entails an attestation statement with its associated certificate information, which is essential in verifying the HSM and key attributes. AWS CloudHSM relies on managed cloud security models that provide storage for securing the cryptographic key. Azure also provides HSM service via Azure Dedicated HSM, which enables users to manage their keys via a hardware security module that is managed within the cloud.

Key management service. The three popular public cloud service providers (Azure, GCP, and AWS) provide key management services. Customers of cloud platforms use key management services to centrally handle the encryption keys within

the cloud platforms [5]. Cloud users can utilize the provided KMS dashboard to create, delete, rotate, and manage access encryption keys. Identity and access management is a service provided by public cloud providers that typically is integrated with the KMS, enabling granulated distribution and access restriction to the keys used for encryption. The encryption keys can also encrypt databases, buckets, and other cloud-based services, including monitoring logs [5].

According to [5], AWS KMS plays a crucial role in protecting customers' root keys and is responsible for encrypting the keys used to encrypt data. AWS KMS also provides key policies that ensure only authorized users have access to KMS keys. Google's KMS lets its cloud customers generate and manage cryptographic keys that are compatible with Google's cloud services [20]. Google's KMS allows users to link their external keys via the EKM (external key management) system. Some other functions of the KMS service provided by Google include provisioning automation via Google's KMS key, performing encryption-decryption activities, and verifying or creating digital signatures used for authentication. Azure provides encryption services via encryption keys that can either be managed by the customer or the cloud platform, a similar trend in AWS and Google Cloud [22]. Azure's KMS can be categorized into key vault and HSM services. Azure Key Vault as a key management cloud service allows customers to encrypt keys as well as sensitive information such as passwords, which use keys stored in HSMs [22].

Every AWS KMS key created costs $1 per month, billed hourly. This applies to both symmetric and asymmetric keys. Automated key rotation costs $1 per key. AWS customers incur no costs regarding AWS-managed keys. There are no fees for customer-managed keys scheduled for deletion unless the deletion is canceled. Additionally, data keys generated by AWS KMS only incur charges for the API call. Azure offers vaults in standard and premium packages, both costing $0.03 per 10,000 secret operations. Key rotation automation costs $1 per scheduled rotation for both packages.

Some of the challenges facing KMS in cloud platforms include increased latency, as encryption operations may slow down. Customer-managed keys regarding client-side encryption may face compatibility issues. Encryption keys need to comply with cloud platform requirements. Non-compliance with FIPS 140-2 Level 3 standards could pose challenges. Regarding client-side encryption, keys may be exposed if the client does not manage them properly.

Misconfigured AWS IAM permissions can lead to serious security issues. Roles or policies that are too permissive might give users or services access they shouldn't have. This can put sensitive data and critical infrastructure at risk. For example, if an attacker has permission to access the iam:PassRole and ec2: RunInstances, they could start an EC2 instance with higher privileges. This can escalate their access and cause unauthorized data access, service disruptions, or even larger security breaches.

IDS evasion and countermeasures. Intrusion detection is significant in a network's security. Attack modifications to help prevent detection by IDS are referred to as IDS Evasion [13]. One countermeasure to IDS evasion is to monitor fragmented traffic closely. In addition, IPS vendors help countermeasure IDS evasion via fragmentation by staying ahead of attackers based on their obfuscation techniques.

Penetration test. A network penetration test, or the ethical hacking of a network, refers to the process whereby hackers utilize illicit methods to find loopholes, weak points, or security systems to bypass the apps within the network [14]. These tests are usually preliminary and are used to root out exposure points that are difficult to identify. They involve several steps, including gathering and analyzing relevant information about a potential client. The second stage focuses on exploring and gaining detailed insights about a network. Step two involves assessing network-based devices' static vulnerabilities by integrating network scanners across the systems and other devices.

Summary. The increase in demand regarding data storage and management has made cloud computing popular, as it provides an alternative solution by storing users' data in the cloud. This increase in storage demand challenges public cloud providers as the storage service needs to be secure, reliable, and user-friendly. Cloud computing relies on the internet for users to migrate data to cloud servers and access them via browser or cloud applications, presenting data security vulnerability during the transfer and cloud storage. The literature revealed that data leakage contributes to about 88% of the challenges faced by cloud data [9]. Additionally, concerns regarding data privacy based on remote use of cloud resources contribute to about 92% of the challenges faced by cloud computing [9]. Data privacy concerns are associated with unauthorized access that might manipulate data during storage or transfer. Data availability also presents a challenge to public cloud providers, who must ensure customers can access computing resources continuously. Vendor lock-in presents a challenge where cloud providers are unable to adapt to customers' needs despite efforts based on improvements and upgrades of resources and services.

These challenges require robust solutions to help address their associated vulnerabilities, with data security as the main challenge. The literature identified solutions such as encryption, certification, and intrusion detection systems. Most studies reviewed by the literature identified encryption as the most effective solution to data security.

The vulnerability associated with data security is based on the current increased demand for DBaaS models, where unauthorized access may be executed via strategies such as SQL injection that expose cloud database data. Cloud databases are interconnected based on different geographical regions to help organizations and businesses reduce latency and improve data transfer performance based on proximity to data centers positioned in multiple geographical areas globally. The interconnections increase the level of vulnerability as the various databases are subjected to cyberattacks. Phishing attacks target cloud customers' login credentials to gain access to sensitive data in cloud platforms. Phishing attacks are achieved via emails and links that may subject users to exposing their login credentials. Lastly, social engineering is also a risk to cloud security, as hackers may act as legitimate agents or customer care to gain access to sensitive information.

These cyberattacks require robust solutions to help seal the security vulnerabilities that expose customers to unauthorized access. The literature identified IDS, encryption, and regular penetration testing as significant in addressing these vulnerabilities. The study focused on encryption due to its effectiveness in data security. The literature presented encryption techniques such as symmetric and asymmetric encryption, which provide layers of protection for cloud data. While symmetric encryption uses a single key, asymmetric encryption utilizes a key pair, enhancing security and simplifying key distribution.

The literature identified the three most popular public cloud service providers, including AWS, Azure, and Google Cloud Platform. These cloud platforms offer KMS, enabling centralized management of encryption keys. KMS can be integrated with IAM, allowing users to control access to encrypted data effectively. These mechanisms help build the trust of cloud users regarding enhanced security measures that help ensure data integrity.

3 PROBLEM

Public cloud providers allow their customers to use different cloud applications based on IaaS, PaaS, and SaaS. SaaS has evolved to include DBaaS, providing users

and organizations with a reliable and scalable database management system. Many organizations take advantage of cloud computing benefits, such as reduced costs and remote accessibility, by outsourcing data storage to cloud service providers. The migration of databases from physical storage to cloud-based via the internet presents the issue of safeguarding data. Cloud data stored in storage within the cloud faces security challenges, such as security leaks, which might subject data to breaches or cyberattacks. Data storage over cloud platforms involves data transfer that may be subjected to cyber threats, threatening data confidentiality, integrity, and availability. Solving these problems is crucial to building user trust and ensuring the secure adoption of cloud technologies.

The reviewed literature identified data safety as the main issue in cloud technology. Cloud service providers and users face security challenges during data transfer, storage, and recovery. Cyber threats such as SQL injection attacks, phishing, and social engineering exploit vulnerabilities in cloud systems. Hackers use these vulnerabilities to advance unauthorized access to cloud data. For instance, phishing attacks may lead to unauthorized access to sensitive cloud information such as login credentials.

Additionally, SQL injections may be used to manipulate database queries to expose cloud data. Another vulnerability involves data replication across distributed database systems, further increasing the risk of unauthorized access to the various storage volumes or units. The security risk associated with cloud data storage presents a need for implementing security measures to safeguard cloud data.

Various methods of securing cloud data include IDS (Intrusion Detection Systems), conducting penetration tests, and encryption. The literature reviewed highlighted encryption as the best method to secure cloud data. Encryption relies on secret keys to encrypt data by converting plaintext data into ciphertext. Unauthorized users require the hidden key to decrypt cloud data, providing an improved level of protection. Cloud service providers offer a symmetric type of encryption, requiring a single key for the data encryption process. However, symmetric encryption presents a significant vulnerability where unapproved access to a single key exposes the data.

On the other hand, two cryptographic keys connected via number calculations are used for encrypting and decrypting data in asymmetric authentication type. These two keys, known as a “private key” and a “public key,” allow encryption key distribution to be better and more securely controlled. With asymmetric encryption, the data remains confidential; suppose there is unauthorized access to the public key. A straightforward, proper method is needed to distribute the keys to help improve data security. However, data encryption can incur costs to users since there is a need to ensure that the systems responsible for the encryption have the necessary capacity and that improvements are made regularly [3].

This study focuses on SSE. With SSE, the data encryption happens at its destination by the application that receives it. AWS has emerged as one of the top cloud service providers that provides SSE with AWS KMS keys. The reviewed literature highlighted that KMS allows users to centrally manage their encryption keys within the cloud platforms. The integration of IAM services with the key management service allows granular distribution and access restriction to the encryption keys. This study’s aims regarding performance evaluation of server-side encryption can provide insights into how server-side encryption may be used to secure cloud data. Furthermore, the limitations of server-side encryption will be reviewed to help provide recommendations for improving security measures in Database as a Service.

4 SUGGESTED SOLUTION

According to [11], cloud providers are experiencing an increased demand for cloud data storage and transfer. The increase in demand poses a challenge for cloud providers in providing a secure, reliable, trusted cloud platform solution. [12] states that safeguarding cloud data via effective encryption strategies and key management is paramount. The research focused on data storage in the cloud, an aspect of DBaaS. The proposed solutions focused on server-side encryption via KMS. The solution considered the AWS cloud platform, particularly AWS SSE via AWS KMS. According to [5], KMS allows cloud platform users to centrally handle all of their encryption keys within the cloud. AWS offers a KMS dashboard to create, delete, rotate, and manage access encryption keys.

Identify and access management services are also incorporated to allow access restriction and granular distribution of the encryption keys. The storage bucket used involves Amazon S3. Amazon S3 uses SSE with AWS KMS (SSE-KMS). The security mechanisms in AWS KMS helped the researcher meet encryption-related compliance requirements. The AWS KMS key location was ensured to be similar to the region of the S3 bucket.

The proposed SSE-KMS encryption workflow entailed the following envelope encryption actions: Amazon S3 bucket requests a plaintext data key and a copy encrypted under the specified KMS key. The second step involved generating a data key by AWS KMS, where the key was encrypted, and AWS KMS sent both the plaintext data key and the encrypted data key to the Amazon S3 bucket. Amazon S3 used the data key to encrypt the data. Then, Amazon S3 removes the plaintext key from memory immediately after release. Finally, Amazon S3 stores the encrypted data key as metadata with the encrypted data.

With decryption, AWS KMS and Amazon S3 perform the following actions: Amazon S3 sends the encrypted data key to AWS KMS in a decrypt request. Step two involves AWS KMS decrypting the encrypted data key by using the same KMS key and returning the plaintext data key to Amazon S3. AWS CloudTrail logs can be used to audit the usage of your AWS KMS keys for your SSE-KMS encrypted data. The following are some of the best practices when performing server-side encryption with KMS keys: Authentication: Implement multi-factor authentication (MFA) to enhance user identity verification.

1. Data encryption: Use server-side encryption with AWS KMS keys to secure data at rest and in transit.
2. Data integrity: Employ cryptographic hash functions to verify that data has not been tampered with.
3. Data recovery: Establish robust backup and recovery mechanisms to ensure data availability in case of loss or corruption.
4. User protection: Educate users on best practices for data security, such as recognizing phishing attempts and safeguarding credentials.

The proposed solution addresses the key challenges of data security in DBaaS environments. Encryption with KMS keys is a secure, reliable, and trusted cloud platform solution. The proposed solution helps protect against data theft, data fraud, data loss, and data confidentiality, some of the main challenges cloud computing faces. With the proposed solution, only authorized users are permitted and allowed to have access to encryption/decryption keys. One best practice is to provide minimal access to the encryption/decryption keys. It is significant to note that the

proposed solution uses a symmetric encryption KMS key since Amazon S3 supports only symmetric encryption KMS keys.

5 METHODOLOGY

This study utilizes a hands-on approach to examine the performance of server-side encryption. The AWS platform was used to implement server-side encryption. The data storage will entail the use of Amazon S3, which is a managed cloud storage solution that stores data as objects in a bucket. Objects can be documents, videos, or images. The hands-on simulation relied on the usage of document data type. The first step was to use an AWS KMS dashboard to create the encryption key demo account. The upload of data required the creation of an S3 bucket in a bucket. According to [17], Amazon S3 supports encryption at rest. Amazon S3 encrypts data at the object level as it writes it to disks in its data centers. The decryption occurs when data is accessed.

5.1 AWS KMS hand-on

An AWS demo account was used to create and test the service and examine the effectiveness of AWS server-side encryption.

5.2 AWS KMS

Amazon Web Service console was used to create a KMS key, as shown in Figure 3.

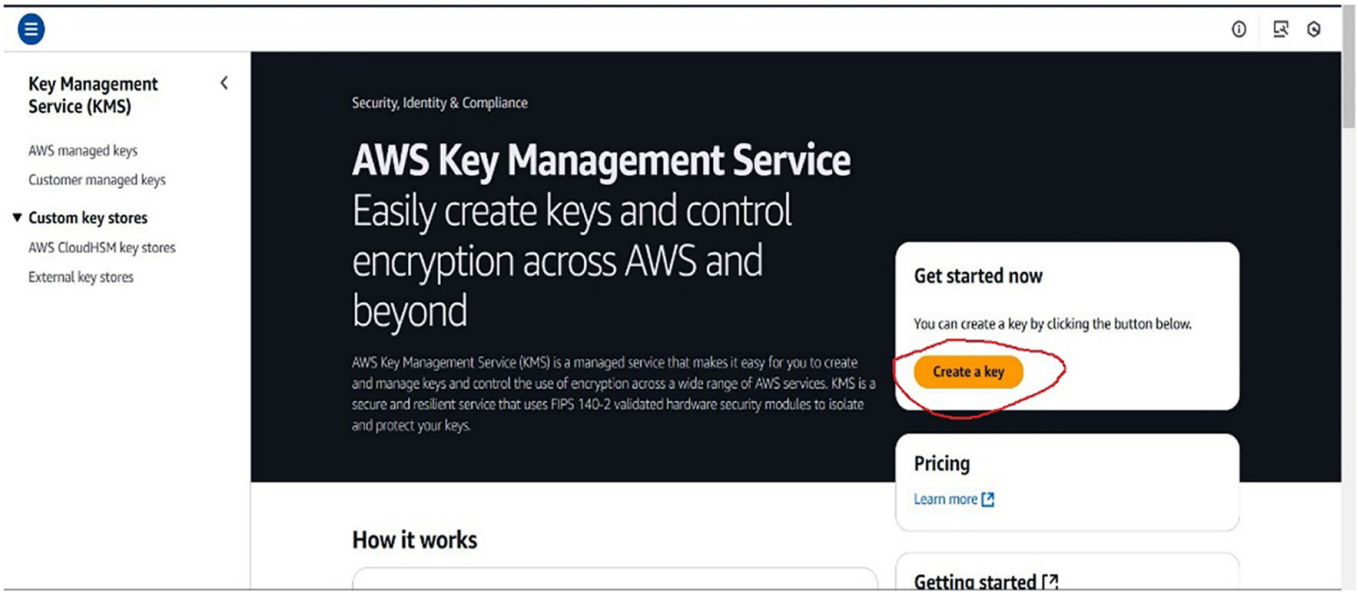

Fig. 3. AWS console KMS dashboard used to create a KWS encryption key

Amazon Web Service SSE-KMS encryption uses a symmetric KMS key since Amazon S3 only supports symmetric KMS keys. The key configuration entailed selecting a Symmetric key type, as shown in Figure 4.

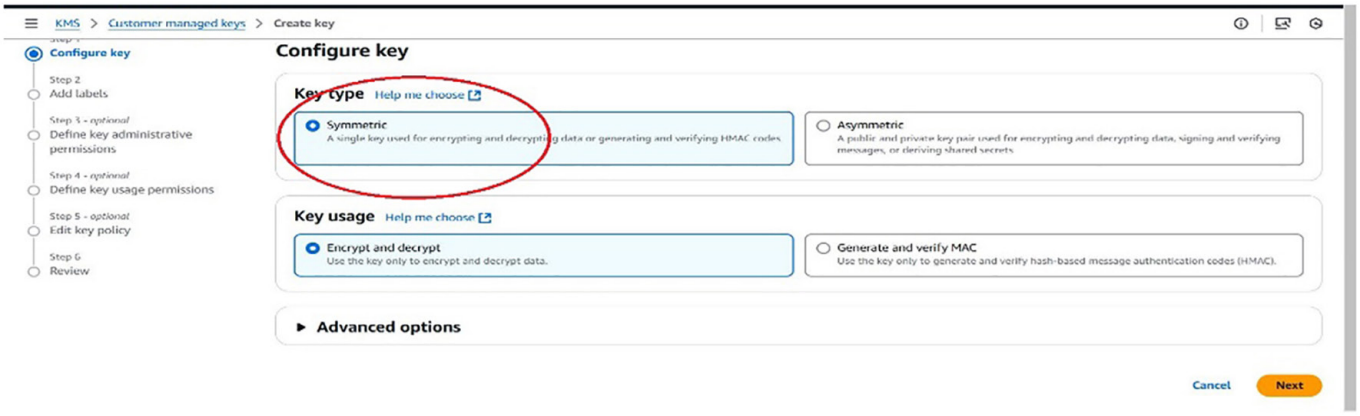

Fig. 4. Symmetric key configuration

An AWS IAM role was configured to help with access restrictions, as shown in Figure 5 below. An IAM role is an AWS identity with permission policies that determine what the identity can do. The service provides the user with temporary security credentials for the role session.

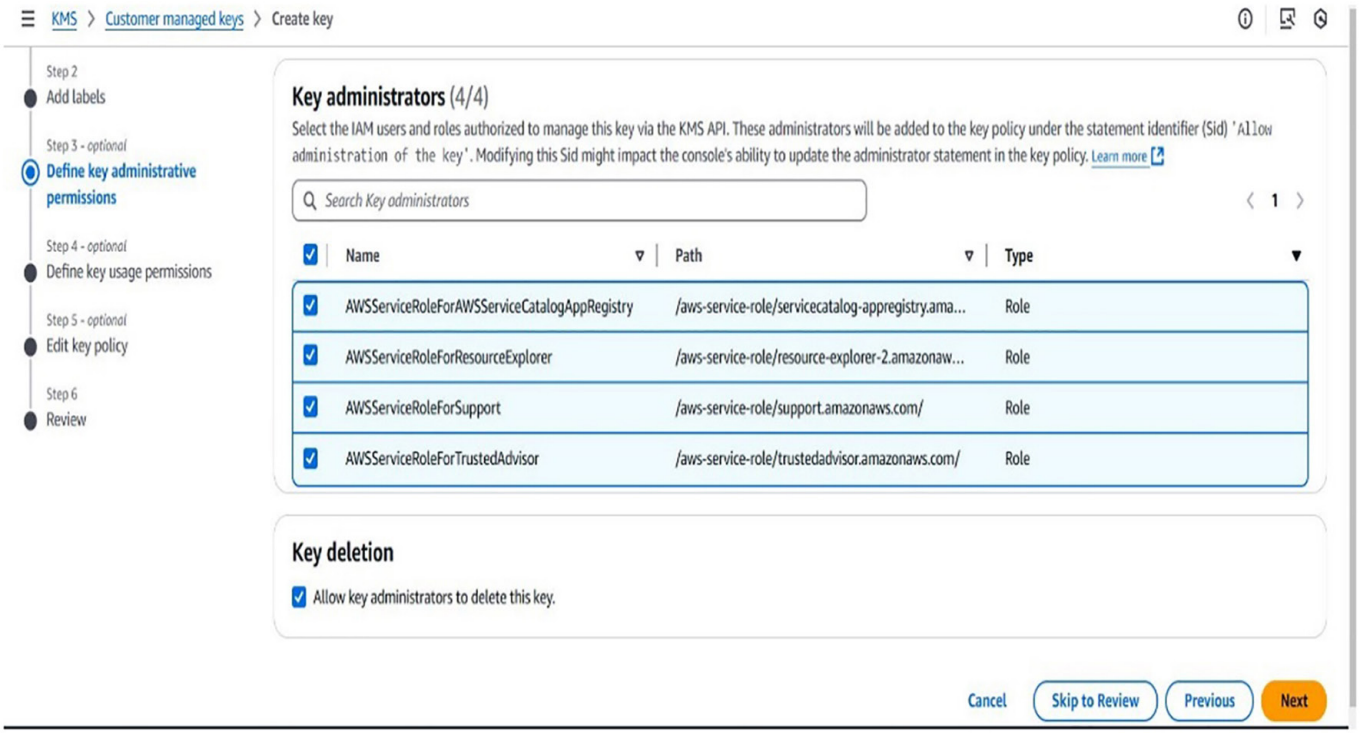

Fig. 5. IAM role configuration

The next step was to configure key usage permission.

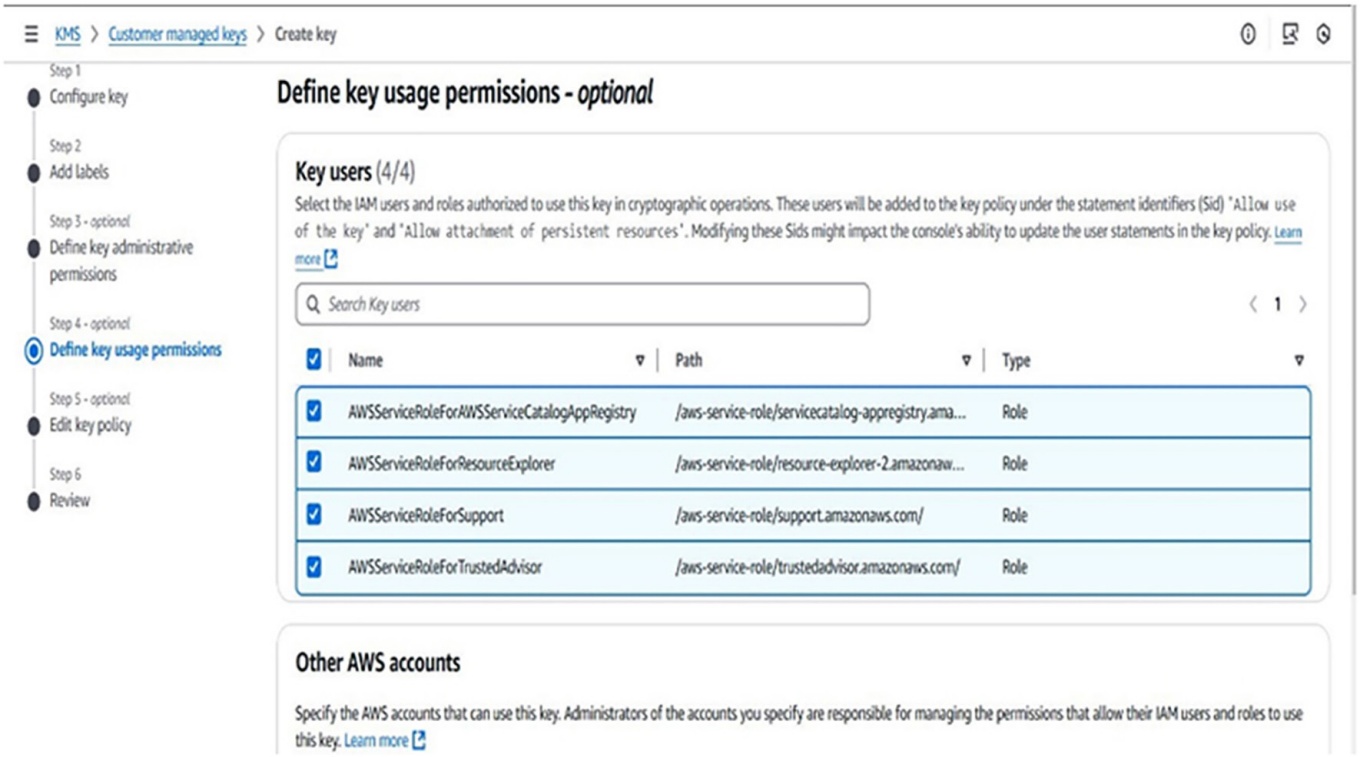

Fig. 6. Configurations of key usage permissions

The next KMS key configuration step was to review and edit the key policy.

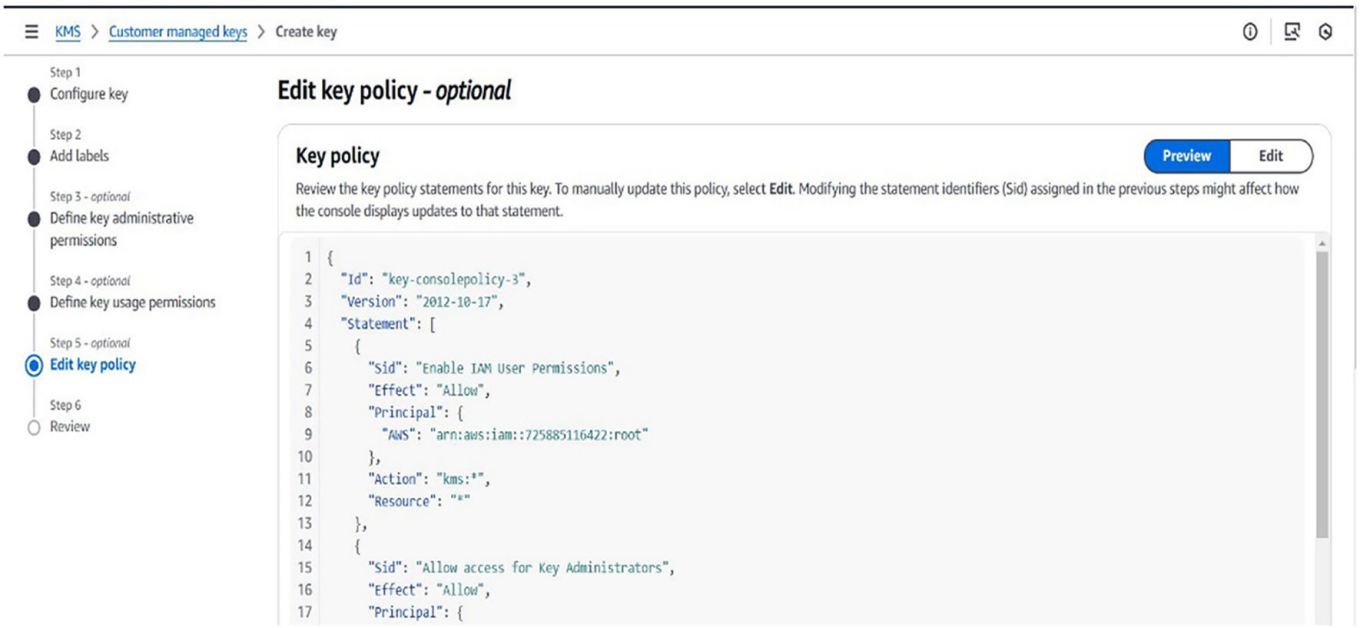

Fig. 7. Key policy configuration

The final step is to review and edit all previous key configurations and create the key, as highlighted in Figure 8.

Fig. 8. Review of configuration steps

5.3 Amazon S3

The next step after KMS key creation entails the creation of an Amazon S3 storage unit. AWS S3, as a managed cloud storage solution, helps users store data as objects in buckets. Buckets are logical containers for objects, and the same geographical region should be used to create Amazon S3 and the AWS KMS. In AWS, the default encryption for the Amazon S3 storage unit is SSE-S3. The researcher used a different encryption method, which entailed SSE-KMS. The AWS console was used to access the Amazon S3 Dashboard and create a storage unit known as a bucket, as highlighted in Figure 9.

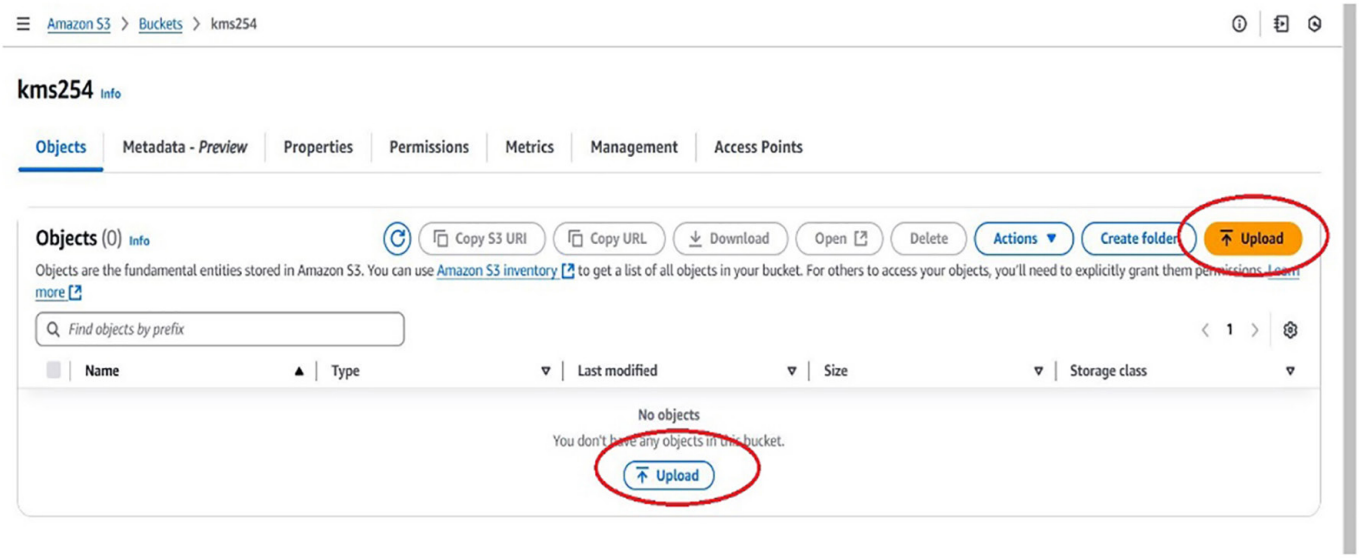

Fig. 9. Amazon S3 bucket creation

A unique bucket name was selected on the bucket configuration page, as shown in Figure 10. Every user must ensure their bucket name is unique across all AWS

accounts and regions within a partition, as one of the requirements by Amazon Web Service.

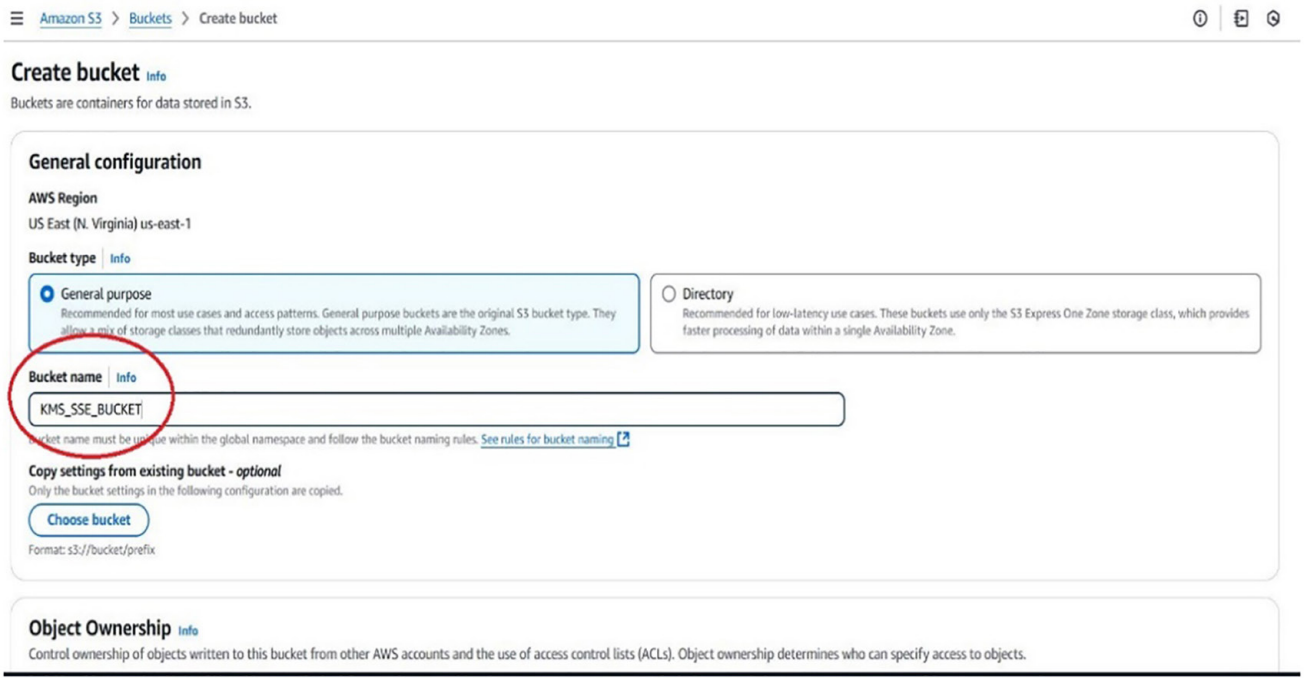

Fig. 10. Unique bucket name selection

The "Server-side encryption with AWS Key Management Service keys (SSE-KMS)" option was selected as the encryption type in the Encryption configuration section. The ARN of the AWS KMS key that was previously created in AWS KMS was used, as shown in Figure 11.

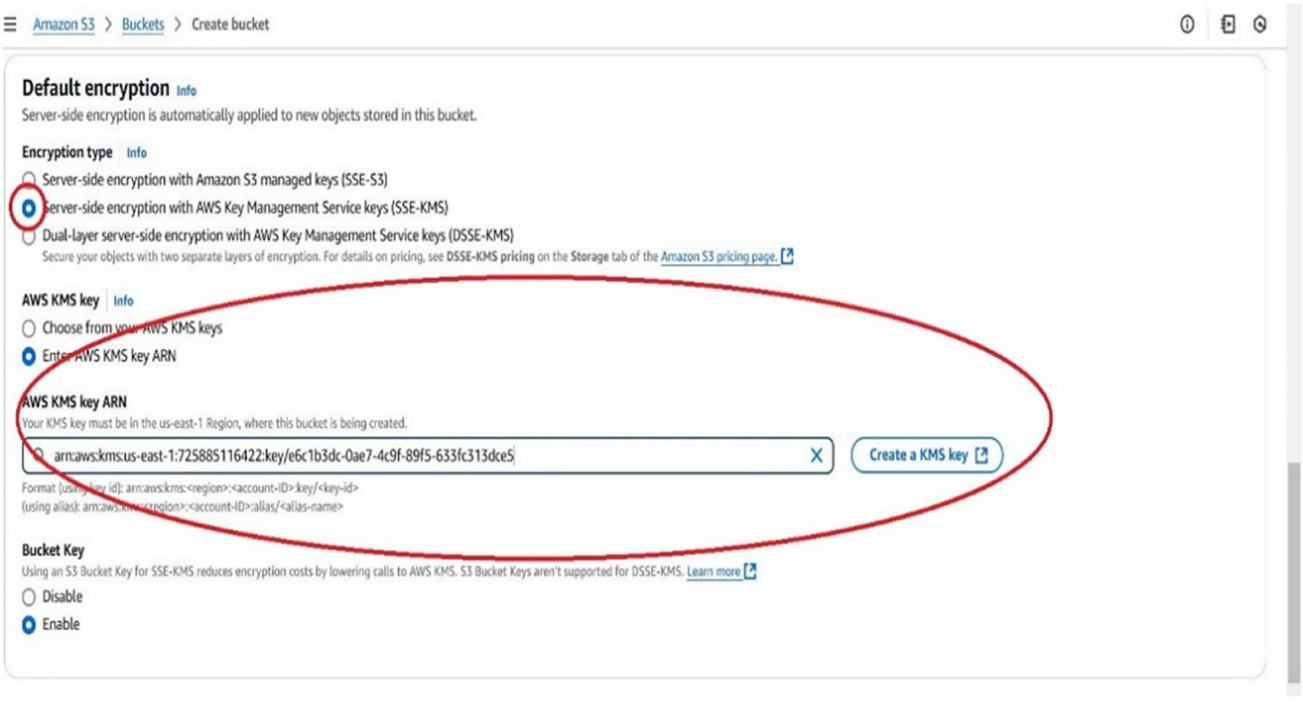

Fig. 11. AWS KMS encryption type selection

The final step is enabling the bucket key option, as shown in Figure 12. Enabling the bucket key option via the dashboard minimizes the incurred encryption costs via lowered calls to the key management system.

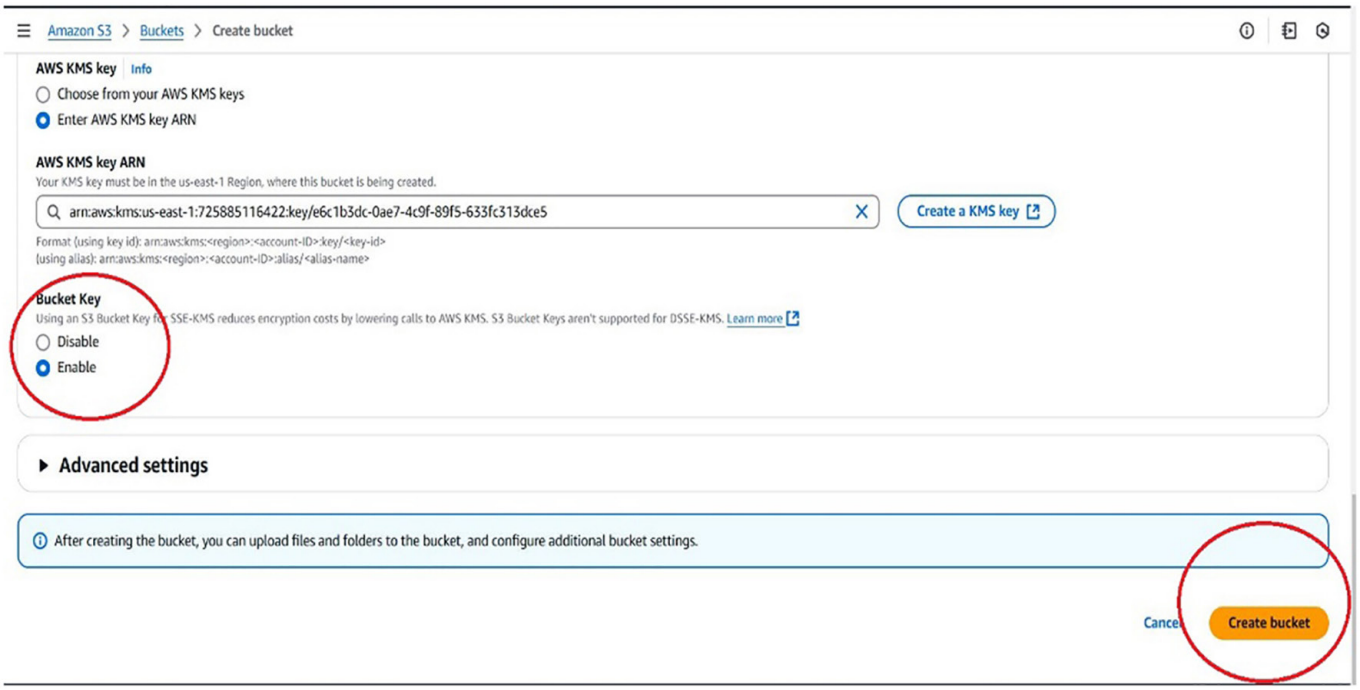

Fig. 12. Selection of bucket key option

After bucket creation, a document file was uploaded via the Amazon S3 dashboard, as shown in Figure 13.

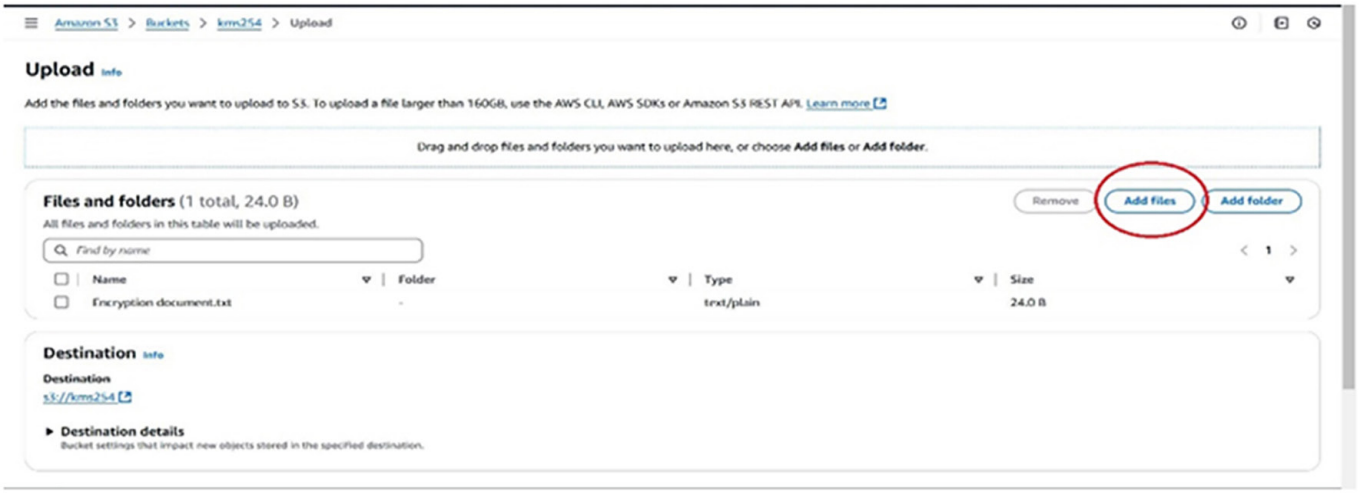

Fig. 13. File upload to Amazon S3 bucket

The browser was used to view the content of the uploaded document; the file was selected and opened via the browser, as shown in Figure 14.

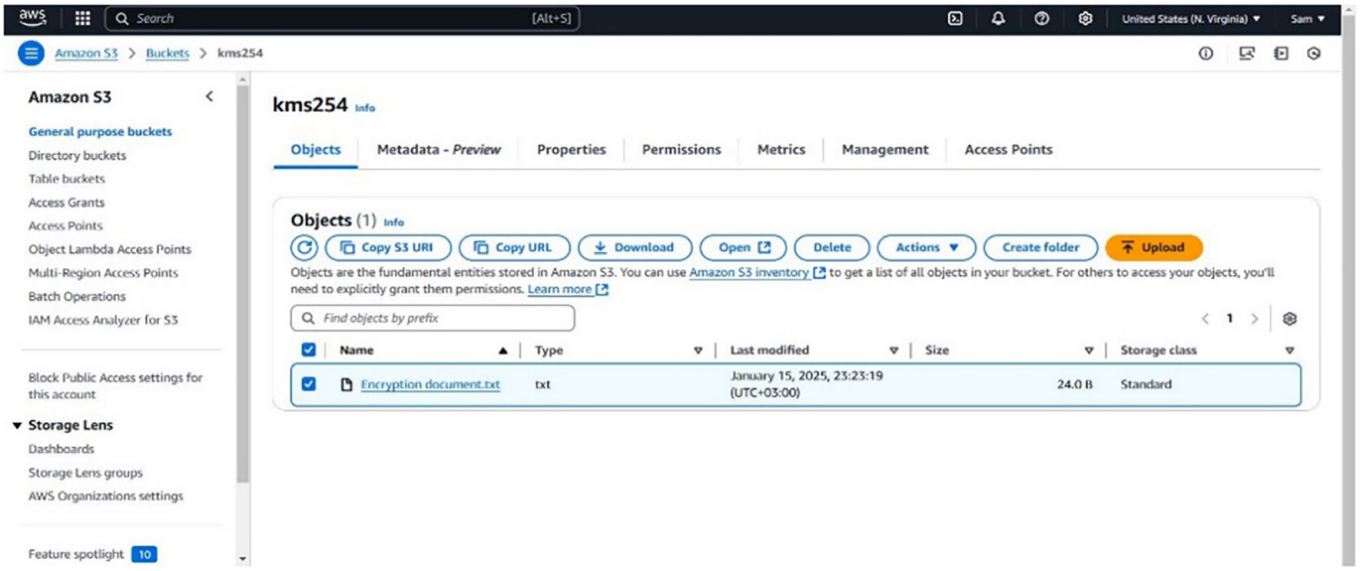

Fig. 14. Content of the uploaded document

The following is the view of the opened file from the authorized user, as shown in Figure 15.

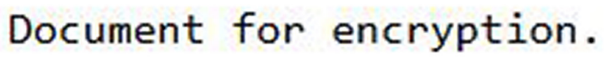

Fig. 15. Document view from an authorized user

The object URL was used to access the bucket object, as shown in Figure 16. The URL below was shared with an unauthorized user.

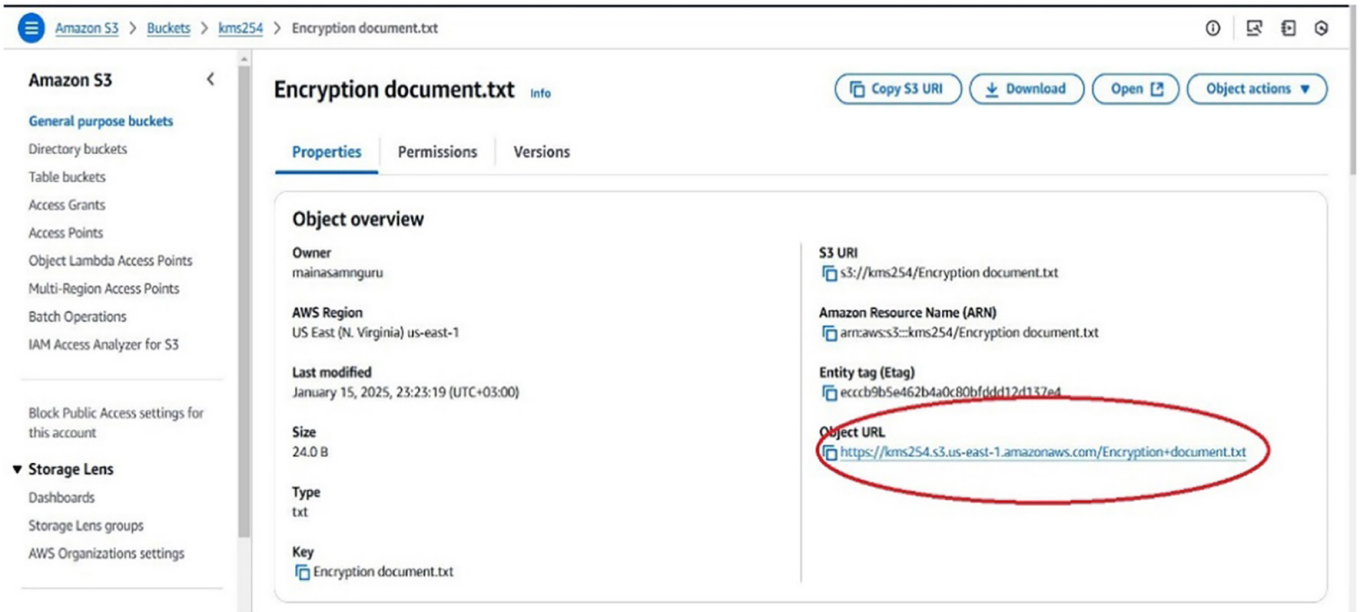

Fig. 16. Object URL retrieval

The unauthorized user accessed the object via the URL retrieved from the previous step. The browser displayed a ciphertext, as shown in Figure 17.

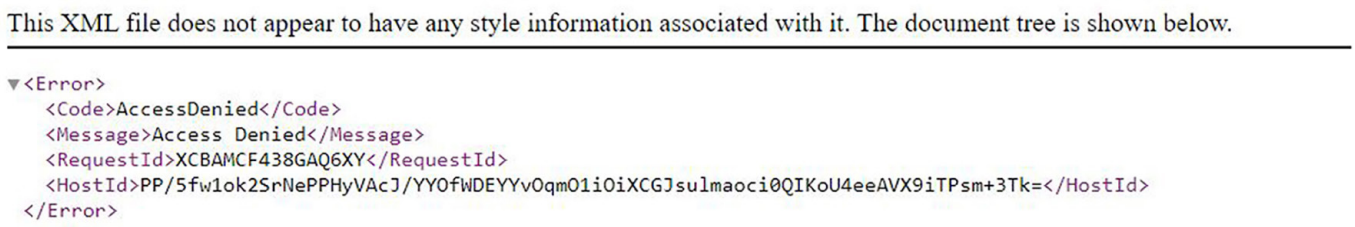

Fig. 17. File view from unauthorized user

5.4 Auditing default encryption

To audit an Amazon S3 bucket's default encryption configuration to ensure AWS KMS encryption was correctly configured. AWS Config was used. AWS Config is a config tool that helps users view, evaluate, and edit the configurations and relationships of the utilized resources. AWS Config has proved to be a valuable tool for determining if buckets are configured to use SSE-KMS or if buckets are unencrypted. The AWS config service was set up as shown in Figure 18.

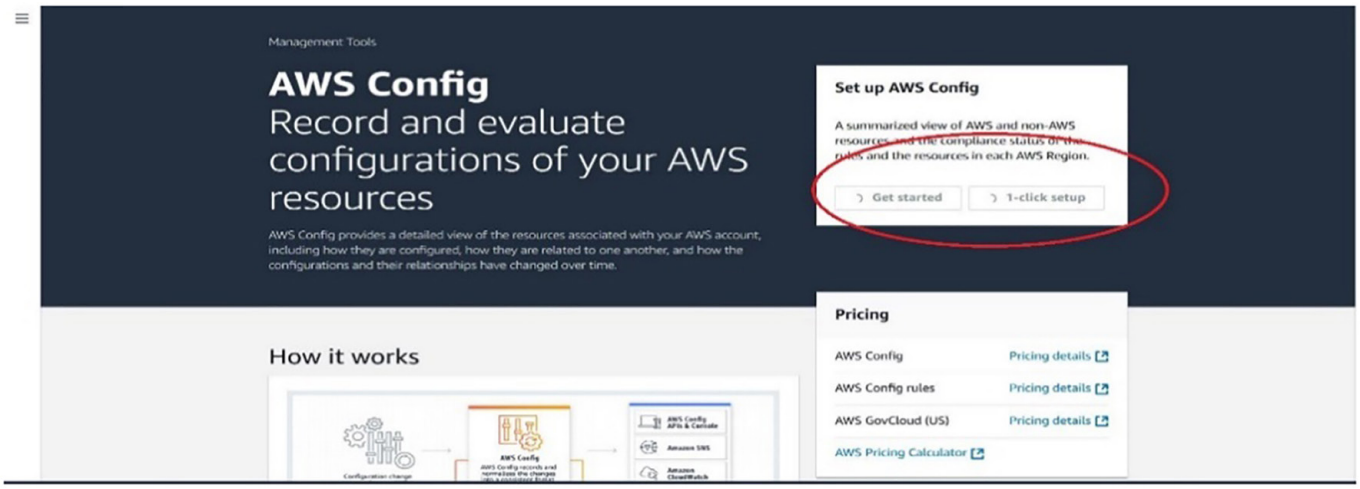

Fig. 18. AWS Config dashboard

The get started button highlighted in Figure 18 directs the user to the AWS Config creation wizard. The confirm button was utilized to complete the AWS Config setup, as shown in Figure 19.

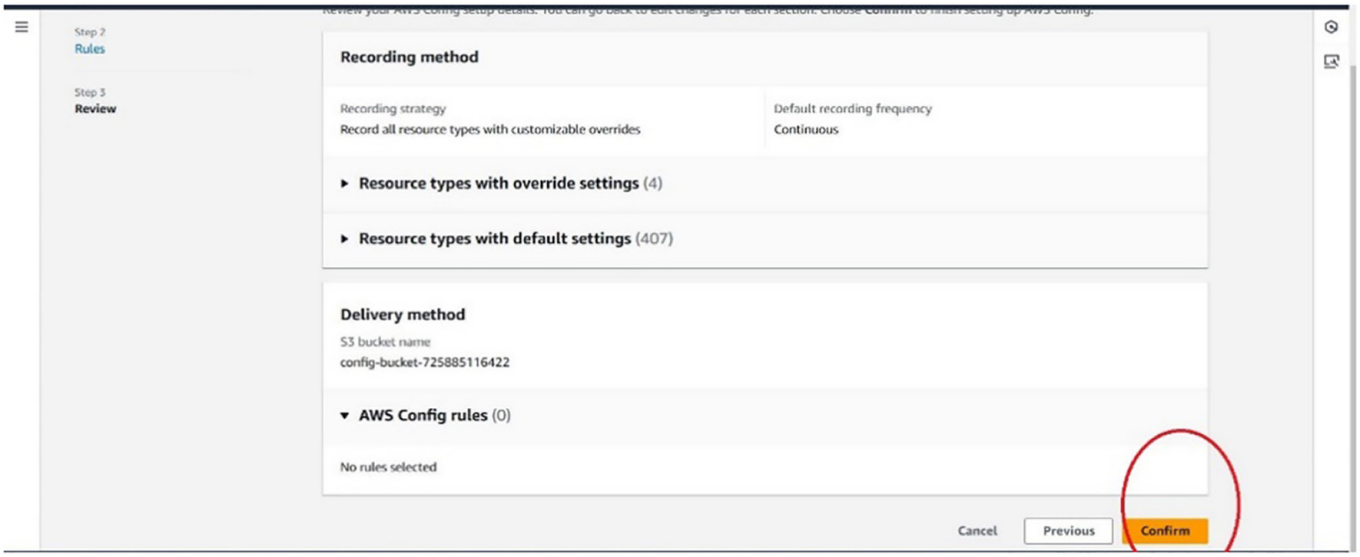

Fig. 19. AWS Config installation Wizard

The resources section was used to access the resource inventory page, as shown in Figure 19. The resource inventory is used to search current resources in use or removed resources, which are recorded by AWS Config. The researcher used the resource details link to view its details. The resource timeline allows the user to view resources based on the time they are utilized. The researcher selected the AWS S3 storage unit used to store the encrypted file, as shown in Figure 20.

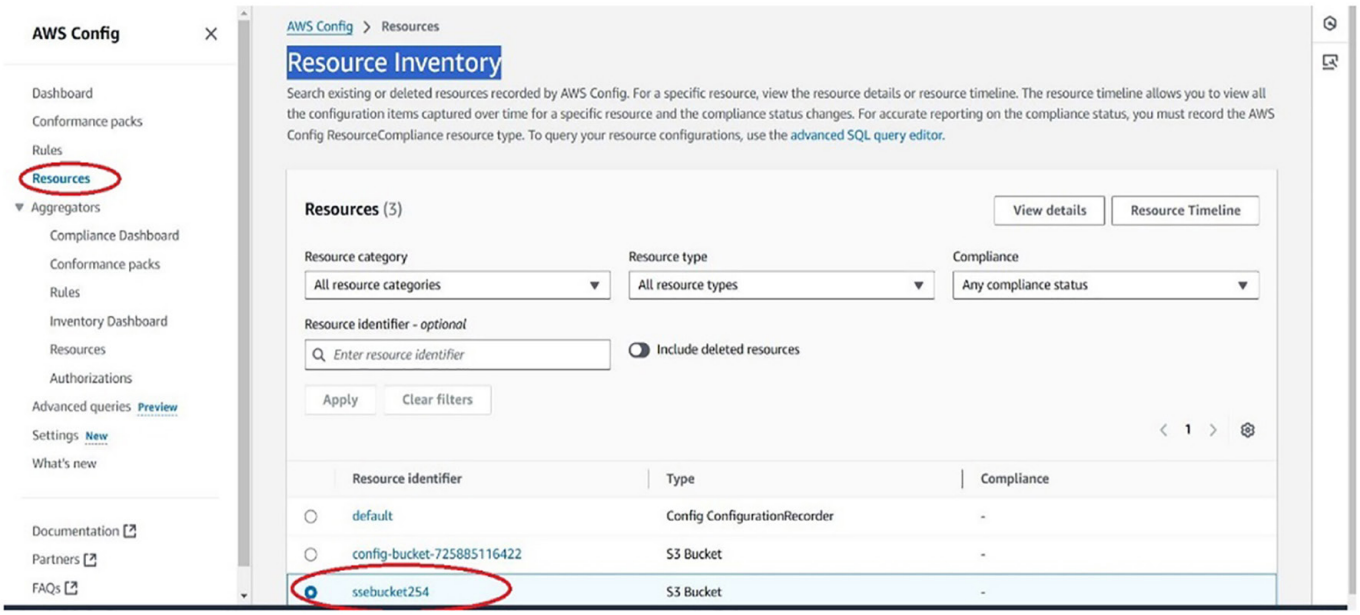

Fig. 20. Resource Inventory feature used to audit

The "view configuration item (JSON)" drop-down section was used to access the Amazon S3 configuration file, as shown in Figure 21.

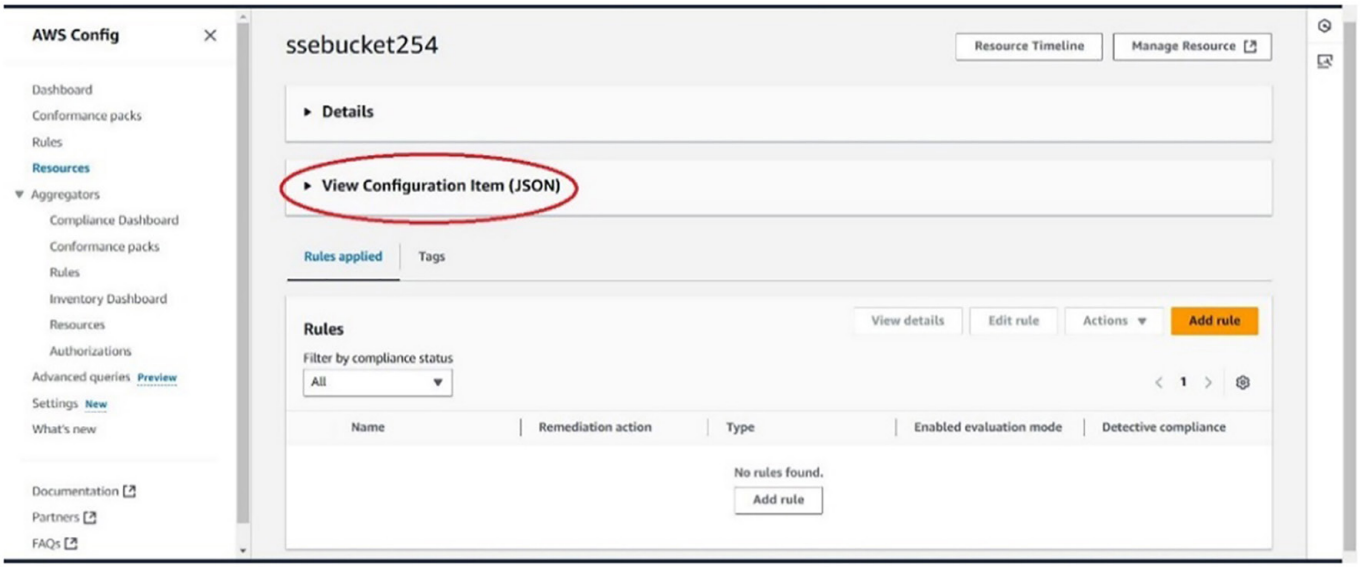

Fig. 21. Amazon S3 configuration file

The opened configuration item was opened. The server-side encryption section showed that AWS KMS was the default server-side encryption, as shown in Figure 22.

Fig. 22. AWS KMS as the default server-side encryption

5.5 Summary

To test AWS server-side encryption, an AWS account was created for analysis purposes, and several tests were conducted. The procedure began from the AWS console, where a KMS symmetric key was created. Since the S3 service supports only symmetric keys, this option was only enabled. Subsequently, an IAM role was created to manage security policies. Specific key permissions usage was then configured, and key policy editing commenced. After ascertaining that all configurations were correct, the key was generated.

The next stage involved the creation of the AWS S3 storage unit with the AWS KMS key. However, it is worth noting that both tools were placed in the same geographical position. As mentioned earlier, Amazon S3 supports only a singular server-side KMS key, so instead of multiple keys, the option was enabled. After the selection of a bucket name, the final task involved turning on the S3 Bucket Key option to utilize and save costs on encryption.

The process begins by creating a bucket, which is followed by uploading a document that can then be accessed by an authorized user in the browser. An unauthorized user, on the other hand, is able to retrieve the object URL but gets cipher text instead of the content of the document.

Lastly, with the help of AWS Config, we could check the default settings for the encryptions on the S3 buckets. This tool was helpful in evaluating whether the bucket had been appropriately set to allow the use of Amazon's KMS service for encryption and how it applied to different configurations. The configuration process was done by accessing the AWS Config dashboard, completing the relevant configurations, and checking the S3 settings from the resource inventory. The resulting configuration item confirmed that AWS KMS was the default Amazon S3 server-side encryption.

6 RESULTS

The results were based on Hands-On, which was performed via the AWS account. AWS provides customers with a KMS that allows the researcher to centrally handle the encryption keys in the cloud platform. This section presents the various configurations used to achieve the results. Validation of the results was performed via AWS config, which confirmed the default SSE was the Amazon Web Service key management system.

The researcher used AWS KMS to generate successful encryption keys. The AWS console KMS dashboard was used to create the KMS keys. The AWS KMS key was a symmetric encryption key. The KMS used other services, such as IAM, that helped ensure authorized users could view the cloud data. Regarding IAM roles, permission for users, resources, and user groups is attached to the Amazon S3 bucket's policies. Access to the created KMS key is attained via the key policies. The researcher retrieved the bucket URL and shared it with an unauthorized user who viewed the cipher data rather than the plain text. The authorized user was able to view the plain text when the bucket was accessed via the AWS console.

The researcher followed a systematic approach to create and configure a KMS key:

Key type selection: During KMS key creation, the researcher selected the symmetric key type. The symmetric keys helped with encrypting and decrypting data stored in Amazon's S3 bucket. The key type was compatible with Amazon S3.

Identify and access management role configuration: During KMS key configuration, an IAM role was configured to allow authorized users and restrict unauthorized users' access to the data. IAM roles defined permissions and provided temporary security credentials for the session, ensuring that only authorized users could use the key.

Key usage permissions: Permissions for key usage were carefully configured to determine which users or roles could manage the encryption keys. Key usage permissions helped improve security by preventing unauthorized access to encryption key management.

Key policy review: The key policy was reviewed, and modifications were made to ensure certain security requirements were met. This step served as the final step of key creation configuration prior to the final submission of the configurations to authorize the key creation.

Amazon S3 bucket creation and configuration

Upon successful completion of KMS key creation, the researcher proceeded to establish an AWS S3 bucket, which was used as a storage facility for the uploaded data. Amazon S3, as an AWS-managed service, stores objects in buckets.

Bucket creation: The geographical region used to create the KMS key was the same region used to create the bucket. Using the same region ensured compliance with AWS requirements based on optimal performance. In addition, the bucket name was made unique, which is another requirement from AWS.

Encryption configuration: During the bucket configuration setup, the selected encryption type was "Server-side encryption with AWS Key Management Service keys (SSE-KMS)." In order to utilize the KMS keys created from the previous section, the KMS key ARN was used in the bucket's encryption configuration setup.

Bucket key option: The researcher opted to enable the Amazon S3 Bucket Key option as it reduces encryption costs based on calls to AWS key management service.

File upload: Upon successful completion of bucket creation, the researcher used the console to access the Amazon S3 dashboard and uploaded a test document to validate that the encryption process was successful. The file upload was encrypted via SSE-KMS as the default encryption for the AWS S3 storage unit.

Authorized access: To help examine the security of uploaded data, authorized users accessed the file and were able to view the plain text, implying the effectiveness of SSE-key management service.

Unauthorized access: To examine the security of cloud data associated with unauthorized access, the file uploaded (object) URL was retrieved by the authorized users and shared as a link to the unauthorized user. The unauthorized user attempted to access the uploaded document (object) via the browser but could only view the ciphertext rather than the plain text. The researcher had allowed all public access to the bucket to help examine its security features. This test showed that SSE-KMS is significant in protecting cloud data.

Auditing: AWS Config was utilized to thoroughly audit the default type of encryption configuration of the Amazon S3 storage unit. This powerful tool offers an in-depth view of the resources linked to your AWS account, detailing their configurations, interrelationships, and the changes in these configurations over time. Upon review, AWS Config confirmed that the default type of encryption for the created Amazon S3 storage unit was set to AWS KMS. This validation demonstrates that the AWS KMS server-side encryption was properly configured, ensuring the secure encryption of the uploaded file.

The researcher selected AWS as the public service provider to help perform the simulation. The simulation entailed the usage of AWS KMS to generate the encryption keys. The encryption type was symmetric, as the KMS keys used were based on symmetric algorithms. The hands-on entailed using an AWS account, and a single region was used to create all services used in the simulation. The researcher used the AWS console to generate the KMS keys and create the AWS S3 storage unit responsible for storing the uploaded data. Validation of the default encryption of the storage bucket was performed via AWS config.

The first step was to use the console to create the KMS encryption keys. During integration, the configuration of KMS keys entailed integrating with configured IAM roles to manage user access and establish permissions based on policies attached to

the encryption type. A final review of the key policy was conducted to ensure the setup was configured based on the requirements.

After the KMS key creation, the researcher proceeded to create the AWS S3 storage unit. The AWS S3 storage unit was made in the same region as the established KMS key. The storage unit was set up to use SSE with AWS-KMS keys. The researcher enabled the Bucket Key option, which helped reduce overall encryption costs based on low calls to the KMS service. After bucket creation, the user uploaded the document to complete the encryption process. Authorized users could view the uploaded document in plain text via the browser, confirming SSE-KMS's effectiveness. AWS config was used to validate that SSE-KMS encryption was the default encryption for the bucket with an uploaded document.

The researcher retrieved the bucket URL and shared it with an unauthorized user. The unauthorized user accessed the bucket with the encrypted file and only viewed the ciphertext of the document rather than plain text via the browser, confirming SSE-KMS's effectiveness.

7 DISCUSSION

The results supported the literature regarding the importance of encryption in addressing security challenges faced by cloud computing in regard to data protection. The results highlighted the successful completion of KMS creation in the cloud platform, allowing the user to manage the keys centrally within the cloud environment. This helped improve performance in terms of the reliability of the encryption key management and enhanced the security of cloud data. The literature revealed the types of encryptions based on SSE, end-to-end encryption, and client-side encryption, which are widely adopted by many organizations and users worldwide. The results from this study demonstrated the significance of server-side encryption, including its benefits in terms of its efficiency and reliability, with a focus on the symmetric type of encryption for data at rest (Amazon S3 cloud data).

According to [4], the computing overhead in symmetric encryption is lower than asymmetric. In other words, symmetric encryption achieved via cryptography is much faster. The improved performance is based on the simpler basic building blocks with lower computational complexity. Low computation overhead is used to ensure the integrity of the message received.

The study results highlighted these benefits regarding symmetric authentication. The usage of KMS as a symmetric type of authentication with integration with a bucket key helped reduce costs incurred by using the encryption service. The integration of IAM role services with KMS helped with the management of restricted access to the encryption keys, establishing an additional layer of security. The access feature was made more effective when resource-based IAM policies and identity-based permissions were configured with KMS key policies. The integration of these policies provided a multi-layer cloud data protection from unauthorized access, providing some of the best practices regarding cloud security measures.

The configuration of the Amazon S3 bucket entailed integrating tighter security measures by adding server-side encryption via the usage of KMS keys (SSE-KMS) created earlier and associating it with the bucket responsible for storing the uploaded data object. The integration of encryption type based on created KMS keys demonstrates how cloud providers have services that help with securing users' cloud data. The AWS simulation revealed that only authorized users could access the uploaded data as plain text. Any attempt at unauthorized access to the bucket revealed a

ciphertext form of the uploaded data. Additionally, the access control mechanism provided by IAM roles and policies helped provide multi-layer protection, proving it difficult for unauthorized users to view the stored cloud data.

The simulation process highlighted the various security configurations based on encryption IAM roles, as well as policies that helped demonstrate the efficacy of server-side encryption. According to [4], symmetric encryption uses AES-256. The algorithm is a symmetric block cipher, which is utilized by governments, including that of the US, to secure sensitive data. Today, the AES encryption standard has become an industry standard for encrypting information, and all popular public cloud providers, including AWS, Azure, and GCP, use the algorithm as symmetric encryption. According to [4], AES-256 encryption is considered the most secure algorithm when compared to other algorithms and is used by the military as well as governments to secure sensitive information. The literature revealed the migration of data to cloud servers by organizations and customers is faced with the challenge of a lack of trust in whether their data will remain confidential when at rest in the cloud storage unit. Cloud data can be vulnerable to unauthorized access via computational attacks. The results demonstrated that the use of KMS with the integration of IAM roles is ideal for mitigating these security risks associated with data breaches and unauthorized access.

On the other hand, the literature highlighted the vulnerability of a symmetric type of encryption, which depends on a single encryption and decryption key for the encryption and decryption operations. The need of both the sender and receiver to use one key poses a challenge to secure key distribution, particularly for organizations with large numbers of users. The results demonstrate that if authorized users expose the cloud login credentials, the cloud data is at risk, as the unauthorized user may use the login credentials to access the cloud data.

According to [3], data encryption can incur costs to users since there is a need to ensure systems responsible for the encryption feature stay updated and improvements are made. However, the AWS cloud provider offers a Bucket Key option, which helps lower the overall costs of encryption services as it enables caching encryption keys locally; this feature minimizes the frequency of calls to KMS. The results demonstrated how a symmetric encryption mechanism can be configured to provide a balance between security and operational performance. Most studies from the literature reviewed revealed that cloud computing helps lower costs for users in cloud computing platforms who need to pay only for the computing resources utilized based on how much they use. With SSE and KMS being lightweight encryption algorithms, users tend to spend less when they adapt this encryption type to secure cloud data.

The literature highlighted centralized key management systems as essential for ensuring compliance with users' needs and organizational policies while still maintaining security at a high level. The results supported this notion by demonstrating that the usage of KMS and IAM roles provided users with the capability to create, manage, edit, and delete encryption keys effectively in a secure manner. These frameworks provide an effective solution for securing sensitive cloud data.

The results and the literature both highlighted how KMS simplifies users' roles in managing keys and effectively securing their cloud data without incurring too much cost. The study results are a guide for users and organizations to adapt to server-side encryption techniques without reliance on complex steps that deter users from considering encryption methods as a way to secure data. The literature revealed that the encryption method is gaining popularity, and there is a need to have a framework in place that can be used to adapt the encryption mechanisms to cloud processes for cloud customers.

8 CONCLUSION

Database as a Service has gained momentum in the SaaS field, where organizations and users outsource their data to cloud service providers, who are responsible for managing and taking control of the data. Encryption strategies have emerged as one of the best strategies for securing cloud data. Encryption methods are ideal for overcoming the cloud limitations associated with loss of confidentiality and strengthening the migration of data to the cloud with a focus on databases as a service. The study focused on SSE via the use of KMS to secure cloud-stored data. The insights gained via simulation over the AWS cloud platform helped provide valuable guidance on KMS encryption best practices that can be used across multiple public cloud platforms.

Encryption is considered the best solution to cloud data security, as it secures data by converting plaintext into ciphertext via encryption and decryption keys. Cloud service providers offer SSE, which enables data to be encrypted at its destination via the service or application at the end. The study used server-side encryption via the integration of KMS keys (SSE-KMS). This encryption approach was achieved via the integration of the AWS S3 storage unit with an AWS KMS service. The integration provided better management of the encryption keys. The AWS SSE with AWS KMS keys (SSE-KMS) simulation enabled the researcher to view the various encryption keys and edit control policies. Therefore, using KMS encryption provides users with centralized key management capability, simplifying the encryption process for users to secure cloud data.

The integration of IAM roles during KMS configurations helps restrict access to the encryption keys based on authorized users or applications. IAM roles are associated with permissions, which are responsible for the authorization of restricted access. Organizations and users can use these mechanisms to minimize the risk of unauthorized access. The KMS encryption service provides an additional layer where data remains confidential, as unauthorized access will view ciphertext rather than plaintext.

The storage services offered by cloud service providers are based on pay-as-you-go pricing. The study findings highlighted that server-side encryption via KMS keys is a symmetric type of encryption, which is lightweight when compared to other encryption methods. [4] states that the computing overhead in symmetric encryption is lower than asymmetric. The low computational resources used will help reduce costs, as pay-as-you-go pricing is based on the quantity of utilized resources as well as the period. The simpler algorithms used in symmetric encryption make server-side encryption with KMS perform faster than asymmetric encryption methods. The seamless performance and low costs make the encryption solutions ideal for many customers with specific needs.

Cloud providers such as AWS offer a management console that users can use to access KMS. The KMS dashboard allows users to centrally handle KMS keys via creation, editing, and deletion features. Users can also establish the policies that control users' and services' permissions. Additionally, users can use the KMS API to audit usage and ensure that the encryption services are being used correctly.

In conclusion, this study has shown that server-side encryption via KMS has simplified the management of encryption keys securely. Access control was further achieved via the integration of IAM roles, which help structure permissions regarding resources and user access restrictions to the encryption service. Additionally, server-side encryption via KMS keys helped strike a balance of cost and performance, which many customers may find ideal. Most cloud service providers offer

KMS, which customers may use to reduce the complexity of using encryption as a method to secure data.

9 FUTURE WORK

The study highlighted the benefits of server-side encryption via KMS keys as a symmetric encryption type. Future research may focus on limitations identified by the study regarding ways to reduce vulnerabilities associated with using a single key for both encryption and decryption in symmetric encryption processes. The hybrid encryption model also presents another research area to help integrate the benefits of both symmetric and asymmetric encryption.

Artificial intelligence (AI) has also gained traction in several fields, such as general intelligent systems, robotics, and language processing. Machine learning (ML) and AI may be utilized to examine risk patterns associated with the use of encryption keys. Future research may focus on the integration of AI into encryption methods to improve the current algorithms and processes.

This study focused on the encryption of data at rest. Future researchers may focus on securing data in transit in cloud environments. Migrating data exposes security vulnerabilities associated with data interception. The literature revealed encryption techniques are significant in securing the transfer of data. Securing migration data helps ensure the confidentiality of data and protection from unauthorized access during the transfer.

Additionally, future research may focus on the integration of an IDS, "Intrusion Detection System," with encryption methods to help detect unauthorized access in real-time. According to the NIST (National Institute of Standards and Technology), IDS refers to software that looks for suspicious activity and alerts [18]. Network IDS may be used to analyze network packets through anomaly approaches to detect any incidence of malicious traffic or activities [18].

One of the benefits of symmetric encryption is that it needs low overhead based on its simpler algorithms. According to [8], integration of digital signatures as authentication mechanisms may help address the man-in-the-middle attack challenge faced by symmetric encryption authentication. However, digital signatures as public key are associated with high overhead. Future research should focus on how to use digital signatures with symmetric keys to address the overhead challenge faced by digital signatures.

10 REFERENCES

[1] D. Salman and N. Sulaiman, "A review of encryption algorithms for enhancing data security in cloud computing," *AlKadhim Journal for Computer Science*, vol. 2, no. 1, pp. 53–71, 2024. https://doi.org/10.61710/kjcs.v2i1.68

[2] P. Modisane and O. Jokonya, "Evaluating the benefits of cloud computing in small, medium, and micro-sized enterprises (SMMEs)," *Procedia Computer Science,* vol. 181, pp. 784–792, 2021. https://doi.org/10.1016/j.procs.2021.01.231

[3] M. B. Qureshi *et al.*, "Encryption techniques for smart systems data security offloaded to the cloud," *Symmetry,* vol. 14, no. 4, p. 695, 2022. https://doi.org/10.3390/sym14040695

[4] T. L. Moore, S. S. Conlon, A. U. Hewarathna, T. B. M. Dissanayaka, and A. B. Mailewa, "Encryption methods and key management services for secure cloud computing: A review," in *Midwest Instruction and Computing Symposium – 2023 (MICS-2023)*, University of Northern Iowa, Cedar Falls, Iowa, USA, 2023.

[5] AWS, "AWS key management service," 2024. https://docs.aws.amazon.com/kms/latest/developerguide/overview.html

[6] V. K. Waghmare and A. K. Kudlikar, "Cryptography challenges of cloud computing for e-government services," *International Journal of Innovative Research in Engineering*, vol. 4, no. 2, pp. 184–192, 2023.

[7] Y. Yan, "The overview of Elliptic Curve Cryptography (ECC)," *Journal of Physics: Conference Series*, vol. 2386, p. 012019, 2022. https://doi.org/10.1088/1742-6596/2386/1/012019

[8] M. Kara, A. Laouid, M. AlShaikh, A. Bounceur, and M. Hammoudeh, "Secure key exchange against man-in-the-middle attack: Modified diffie-hellman protocol," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika,* vol. 7, no. 3, pp. 380–387, 2021. https://doi.org/10.26555/jiteki.v7i3.22210

[9] I. Zulifqar, S. Anayat, and I. Kharal, "A review of data security challenges and their solutions in cloud computing," *International Journal of Information Engineering and Electronic Business (IJIEEB)*, vol. 13, no. 3, pp. 30–38, 2021. https://doi.org/10.5815/ijieeb.2021.03.04

[10] Anjana and Ajit Singh, "Encryption algorithms for information security in cloud computing: A detailed study and analysis," *Migration Letters*, vol. 21, no. S7, pp. 1100–1109, 2024.

[11] S. R. Gudimetla, "Data encryption in cloud storage," *International Research Journal of Modernization in Engineering Technology and Science,* vol. 6, pp. 2582–5208, 2024.

[12] M. Blessing, "Cloud encryption strategies and key management," *Computer Security and Reliability*, 2024. [Online]. Available: https://www.researchgate.net/publication/383660212_Cloud_Encryption_Strategies_and_Key_Management

[13] M. M. Khan, "Developing AI-powered intrusion detection system for cloud infrastructure," *J. Artif. Intell. Mach. Learn. & Data Sci.*, vol. 2, no. 1, pp. 1074–1080, 2024. https://doi.org/10.51219/JAIMLD/mohammed-mustafa-khan/255

[14] I. Yurtseven and S. Bagriyanik, "A review of penetration testing and vulnerability assessment in a cloud environment," in *2020 Turkish National Software Engineering Symposium (UYMS)*, 2020, pp. 1–6. https://doi.org/10.1109/UYMS50627.2020.9247071

[15] K. S. Alqahtani, A. M. Albalawi, and M. Frikha, "Reviewing of cybersecurity threats, attacks, and mitigation techniques in cloud computing environment," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 6, pp. 2058–2066, 2023.

[16] S. Alder, "Florida Health Kids Cooperation announces that 2,000 patients were impacted by a phishing scam," *The HIPAA Journal*, 2017. [Online]. Available: https://www.hipaajournal.com/florida-healthy-kids-corporation-announces-2000-patients-impacted-phishing-scam-8974/

[17] AWS, "Amazon S3 encryption client," 2024. [Online]. Available: https://docs.aws.amazon.com/amazon-s3-encryption-client/latest/developerguide/client-server-side.html

[18] National Institute of Standards and Technology, "Security best practices for the electronic transmission of election materials for UOCAVA voters," *NISTIR 7711*, pp. 1–73, 2011. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7711.pdf

[19] M. Sommerhalder, "Hardware security module," in *Trends in Data Protection and Encryption Technologies*, V. Mulder, A. Mermoud, V. Lenders, and B. Tellenbach, Eds., 2023, pp. 83–87. https://doi.org/10.1007/978-3-031-33386-6_16

[20] Google, "Cloud key management service overview," 2025. [Online]. Available: https://cloud.google.com/kms/docs/key-management-service

[21] AWS, "AWS CloudHSM documentation," 2025. [Online]. Available: https://docs.aws.amazon.com/cloudhsm/

[22] Microsoft, "Key management in Azure," 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/fundamentals/key-management

[23] M. N. Alenezi, H. K. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *International Journal of Communication Networks and Information Security*, vol. 12, no. 2, pp. 256–272, 2020.

[24] AWS, "About data encryption," 2025. [Online]. Available: https://docs.aws.amazon.com/prescriptive-guidance/latest/strategy-data-at-rest-encryption/about-data-encryption.html#:~:text=There%20are%20two%20types%20of,for%20large%20amounts%20of%20data

[25] N. Kshetri, M. M. Rahman, M. M. Rana, O. F. Osama, and J. Hutson, "algoTRIC: Symmetric and asymmetric encryption algorithms for Cryptography—A comparative analysis in AI era," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 12, pp. 1–14, 2024. https://doi.org/10.14569/IJACSA.2024.0151201

[26] H. T. Sihotang, S. Efendi, E. M. Zamzami, and H. Mawengkang, "Design and implementation of Rivest Shamir Adleman's (RSA) cryptography algorithm in text file data security," *Journal of Physics: Conference Series*, vol. 1641, no. 1, p. 012042, 2020. https://doi.org/10.1088/1742-6596/1641/1/012042

[27] M. Anas, R. Imam, and F. Anwer, "Elliptic curve cryptography in cloud security: A survey," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022, pp. 112–117. https://doi.org/10.1109/Confluence52989.2022.9734138

11 AUTHOR

Dr. Waleed Almuseelem received a Ph.D. in Computer Science and Technology from the College of Computer Science and Technology at the Wuhan University of Technology, China, in 2017. He received an M.S. in Computer Science and Technology from the Wuhan University of Technology, China in 2012. His research interests are cloud computing security and data privacy. He works as an Associate Professor at the Faculty of Computer and Information Technology at the University of Tabuk, Saudi Arabia. His research interests include cloud computing security and data privacy (E-mail: waleedalmuseelem@ut.edu.sa).