Journal of Information Systems Engineering and Management
2025, 10(18s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Multilevel Encryption for Multiple-Cloud Storage (MEMC) with Bot Detection and Elimination (BDE) in Recommendation Systems

J. Chitra[1], S. K. Piramu Preethika[2]
[1,2] *Department of Information Technology,*
Vels Institute of Science, Technology & Advanced Studies, Chennai, Tamil Nadu, India
[1]*chitrakarthik452@gmail.com.* [2] *piramu.scs@velsuniv.ac.in*

ARTICLE INFO

Received: 18 Dec 2024

Revised: 28 Jan 2025

Accepted: 12 Feb 2025

ABSTRACT

Introduction: In general cloud acts as an intermediate in Publish and Subscribe (pub/sub) systems to broadcast the publisher's data to the subscribers. But a direct connection between the subscriber and the publisher is not advisable, only a loosely coupled manner allowed along with the cloud. There are a lot of chances in data getting exposed to the attackers from the client side, publisher side or even from the cloud side. Attackers, specifically botnet controllers, use stealthy commanding systems to set up large-scale controls. Encryption before upload is the best way of protecting sensitive publications and subscriptions.

Objectives: To suggest a security system using Multi-level Encryption mechanism for multiple cloud storage(MEMC) with BotDetection and Elimination(BDE) strategy for recommendation systems.

Methods: . A privacy-preserving publish and subscribe system that protects both the publications and the subscription's personal details and identities is presented. The cloud storage that compromises the pub and subsystem is broken using a high multi-level encryption approach. A novel paradigm emerges from combining multi-level encryption with multiple cloud storage (MEMC).A 'Searchable Encryption (SE)' technology is used to ensure encrypted publishing keyword matching, in addition to subscriber interest. The system's efficiency is based on employing various cloud storage for matching and directing trustworthy publications to trustworthy interested subscribers. The optimum solution is to divide the match procedure into multiple phases, coupled with encrypted subscriptions and publication tags. • The partitioned data will be handled by various cloud servers to prevent sensitive data leaks. Even if a single cloud server is attacked and collaborates with a subscriber or publisher to leak data, data privacy is unaffected; subscriptions and publications remain protected. The system uses the 'BOT detection and Elimination (BDE)' methodology to detect and stop BOT control.

Results: MEMC clearly outperforms PEKS, SE, and AES in the Confidentiality criteria, with a score above 3.5. This suggests that MEMC is better suited to maintaining data confidentiality. For the Privacy Maintenance metric MEMC significantly excels, with a value close to 4, showing that the method is effective in the maintenance of user privacy with multilevel encryption. The other methods such as PEKS and AES show relatively lower privacy scores, reflecting their inability to present a solution to the enhanced security concerns

Conclusions: The proposed mechanism is used to secure the publications and subscriptions by providing a multilevel encryption technique before it is disseminated to the cloud. To tackle the untrusted cloud the system provides an Epub/Esub encryption system to maintain confidentiality. Further, the BOT type attacks are common in clouds to resist and overcome this system and introduces a BOT detection and elimination technique. Finally, the experimental results ensure the efficiency, performance and feasibility of the system.

Keywords: Mulit-level Encryption, Multiple Cloud Storage, Bot Detection and Elimination(BDE),Publish and subscribe System, Recommendation System

INTRODUCTION

Commonly cloud acts as a supporting storage unit that stores enormous user data and the publisher data. At times there are chances that the user data or the publisher's sensitive data maybe misused or hacked by multiple attacks. Even clouds may get compromised for exposing the sensitive data to the malicious entities. Hence, there is no guarantee for data when it reaches a public commodity. Simultaneously, pub and sub systems work in a very similar way that the data from the publications are recommended according to the interested subscribers in a loosely coupled way. The main form of transmission in this Publish and Subscribe (pub/sub) system is accelerated through recommendation. In general, the data created by the publishers are represented as publications that are referred to the interested subscribers viaservers called cloud.

The cloud services providers offer cloud services as Software as a Service (SaaS) to the entire network. Basically, publication content is defined with a set of tags with a keyword extraction. Accordingly, considering the set of constraints on these tags the interested subscriptions are made by the subscribers. To appraise the interest of the subscribers in a particular publication the set of tags are matched with interests registered by the subscribers. By identifying interest for the specific publication stored in the server, the publication is directed to the intended subscribers automatically. The proposed work deliberates an extension of the preceding work [1] that emphasizes privacy over Publish and Subscribe Systems. Thepublish and subscribe system is used widely in several applications such as healthcare, marketing, stock exchange, book or article publishing etc., [2-4]. Google, which is the real-time messaging service, offers the pub and sub system for analytics and computing systems (event- driven) [5]. Publish/Subscribe systems have security that protects secret data through encryption techniques. Therefore, it ensures that the data is protected both in its transmission and storage. Additionally, access to the data is also prevented, hence maintaining system integrity and reliability.

Equally important is that the access control mechanism robustly prevents unauthorized entities to subscribe or publish a message within the system [6]. Stricter authentication and authorization rules can be used for authentication purposes, thus ensuring legitimate subscribers only have access, with this enhancing the overall security level of the system. Accounting the benefits of pub and sub systems the demerits and challenges cannot be ignored that cause a major impact on privacy of data by a set of clouds. Accounting the outsourcing unit, the pub/sub service relies on the cloud servers which can be compromised easily. Unfortunately, these cloud servers cannot be trusted in 2016 and the yahoo attack caused leakage of 1 billion accounts leading to privacy issues [7].

Some of the existing methodologies researched by the authors are discussed for a proper pub and sub management. W.Rao et al., in [8] produced a mechanism correlating subscribers and cloud servers possessing a secure pub/sub system. A system resisting collusion attacks among the un-trusted subscribers (or publishers) and cloud server are discussed in previous approaches in [9,10]. The system processes direct communication for publishers and subscribers' privacy against colluding parties. A secure broker-less publish/subscribe system projected by Tariq et al., where the publication is processed by a trustable publisher. For encryption the system uses CPABE Ciphertext-Policy Attribute-Based Encryption method in [11]. Also, a public-key encryption technique followed in processing keyword search [12]. F. Hahn and F. Kerschbaum [13] projected a new scheme named SUISE symmetric Searchable Encryption that processes encryption by interests and tags and acknowledges a secure encrypted subscription.

Y. Polyakov et al., [14] stated a secured topic related to the publish/subscribe system related to proxy encrypting technique. Further the author applied a lattice proxy encrypting scheme projecting a homomorphism operation with a loosely coupled property of publish or subscribe systems. Pires et al. [15] currently introduce a routing engine offered by SGX enclaves for pub/sub system leveraging a trustable execution. Shahid et al. (2022) proposed a blockchain-based framework to enhance the security of IoT-enabled Publish/Subscribe systems. Their work highlights the use of blockchain for immutable logging and secure communication, ensuring resistance to tampering and malicious activities [16].

Mahmood et al. (2022) introduced an adaptive access control mechanism leveraging Attribute-Based Encryption (ABE) for IoT Publish/Subscribe systems. Their approach dynamically adjusts access permissions, improving both security and scalability in IoT environments [17]. Lee et al. (2022) developed a machine learning-based model for detecting anomalies in Publish/Subscribe systems. Their method enhances the reliability of such systems by identifying unusual patterns in real time, ensuring robust performance under diverse scenarios [18].

Dinh et al. (2022) proposed a privacy-aware access control scheme for encrypted Publish/Subscribe systems. Their

approach incorporates fine-grained data access policies, addressing security challenges in untrusted environments [19]. Prasad et al. (2023) explored the use of homomorphic encryption to secure cloud based Publish/Subscribe systems. Their work ensures that sensitive data remains encrypted even during computational processes, enhancing privacy and security [20]. Meshram and Rao (2022) presented an efficient method for executing multi-attribute range queries while preserving data privacy in Publish/Subscribe systems. Their technique addresses both performance and privacy challenges in distributed systems [21]. Lin et al. (2023) introduced a resilient and privacy-preserving Publish/Subscribe framework with fine-grained control. Their system is designed to withstand adversarial conditions while maintaining robust security and confidentiality [22].

To secure the publication and subscription from the untrusted entities and to handle the sensitive data some preventive measures and methods are used in the paper. The privacy protection method and attack prevention techniques are deliberated as follows:

- Initially in this paper privacy preserving publish and subscribe systems protecting both the publications and subscription personal details and identities are proposed. The multiple cloud storage compromising the pub and sub system is broken by following a high multi-level encryption technique. The formation of multi-level encryption using multiple cloud storage (MEMC) together forms a new model.

- A 'Searchable Encryption (SE)' technique is followed to assure encrypted publication keyword matching beside the interest of subscribers. The system efficiency relies on using multiple cloud storage for matching and routing the trustable publication to the trustworthy interested subscriber. The ideal thought is to split the match operation to several phases along with encrypted subscriptions and publication tags. Each phase is assigned to a different cloud server for privacy maintenance.

- The partitioned data will be processed by different types of cloud servers to avoid the sensitive data leak. Even if the one cloud server is compromised and colludes with a subscriber or a publisher for data leak it will not affect any data privacythe subscriptions and the publications are still protected.

- Further, to identify and stop the BOT controlling the system introduced 'BOT detection and Elimination (BDE)' model.

The rest of the paper is organized as follows. Section 2 describes the requirement for secure access. Section 3 possesses a discussion on surveys related to previous work. Entering section4 both the proposed system model and the threat model are discussed together. Section 5 provides an experimental security analysis with comparative analysis. Finally, the paper is concluded in Section 6 highlighting some future research directions

METHODS

The proposed approach implementation with two divisions system Approach and the threat approach. Then, an extended overview of the method implementation is processed. This design method of the privacy-preserving publish/subscribe system allows secure private communication of publishers and subscribers against possible threats such as malicious users and untrusted cloud servers. It starts from the publisher's end, which encrypts the publications made as well as the subscriptions made at the subscriber's end. Here, the subscriber created publication on predefined tags that are earlier encrypted and stored in a specialized Cloud Server C. Exactly in the same manner, the subscribers encrypt their interest and forward them to, yet another cloud server known as Cloud Server A, who keeps confidential all about the subscription. The system has a multi-cloud architecture where all the roles are played by each cloud server. The function of encrypting and sending data safely is taken up by Cloud Server B. There is a trusted authority in place, managing and providing encryption keys to publishers as well as subscribers. It is responsible for controlling access to information. Only authorized entities are allowed to interact with the system and retrieve the encrypted data.

A BOT tracker is an important methodology that will monitor and prevent malicious automated activities. It detects suspicious BOT commands or attempts to access the system in unauthorized ways, ensuring the system integrity will not be violated and preventing any intrusion that may compromise the publication and subscription process. It is a strategy that combines techniques of advanced encryption with multiple cloud storage and real time malicious activity monitoring for maintaining privacy, confidentiality, integrity of information in the publish / subscribe communication system. Such a system ensures secure and efficient and resilient handling of secret information that

also addresses issues such as user's privacy and external attacks in decentralized cloud architecture. A privacy-preserving pub/sub service mode is executed following the below steps:

The system model undergoes several sections for publications, Sub section for subscriptions, cloud server (A, B, C) and trustable authority. Figure 1. provides a detailed system approach with the proposed implementation design. A publish/subscribe service providing protection to the Pub and Sub from the curious cloud server and malicious pubs and subs are provided.

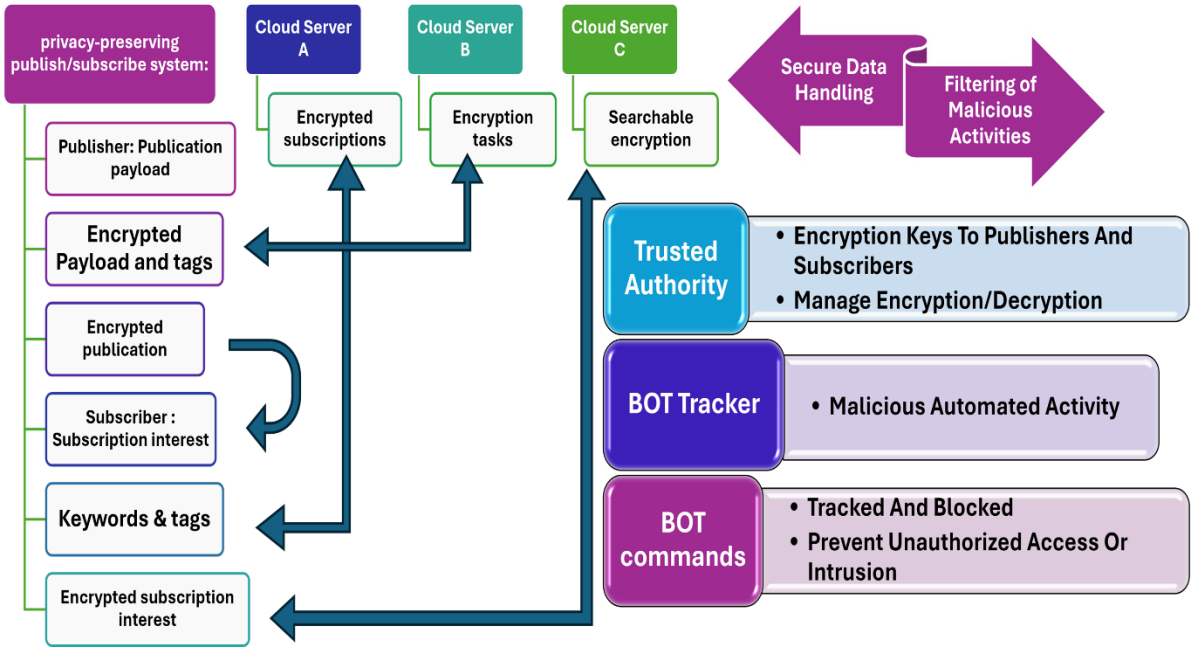

Figure 1. system architecture for proposed Pub/sub system model.

The architecture diagram depicts flow and interaction among the different components in a privacy-preserving publish/subscribe system. It starts off with a depiction that the publisher possesses publications that carry a long payload along with associated tags. These are encrypted and then sent to Cloud Server C. In the same way, subscribers express their subscription interests that again get encrypted and transmitted to Cloud Server A. Other than decrypting subscriptions, Cloud Server B also handles management of encryption and transfer operations and ensures safe processing within the system.

A trusted authority would oversee the key management system for encryption and decryption to provide keys to publishers and subscribers. This keeps the resources of the system secure from people who are not authorized while avoiding letting out sensitive data. It tracks, monitors by the BOT tracker, malicious automated activities or unwanted access to prevent potential intrusion or threat. It takes a major role in monitoring suspicious BOT commands that can inflict damage and block them early.

The system has several cloud servers for handling publication, encrypting, and managing the subscription, which increases security for data and privacy. Since it integrates a BOT tracker in its architecture, the malicious activities cannot interfere with the working of the publication-subscription process. This architecture represents protected communication, that is privacy preservative within a decentralized cloud. This architecture protects against malicious cloud servers, publishers, and even subscribers.

- Encrypted Pub (Publishers) section: In this section the publisher creates publications with their respective tags. For the publisher a specific cloud server is assigned before storing the publication; both the payload and tag of publication are encrypted.
- Encrypted Sub (Subscribers) section: In the Sub section the interests of the subscribers is collected and encrypted subscriptions are made according to their interests. Only thepublications that satisfy the subscription tags are recommended to the subscribers.
- Cloud server: for each section a separate cloud server is allocated. In the system 3 different types of cloud server are assigned cloud A for subscriptions, cloud B for encryptionand transferring, Cloud C for publications. The

cloud server performs the filtration anddelivering of publications according to the subscribers' interest.

- Trustable encrypted Authority: this section handles the key authorization to manage thesubs/pubs system keys.

EMC with SE Implementation*:*

BOTS is a term for robots; it is an automatic system of software or scripts doing things over the Internet, for example, data gathering and interaction with services performing repetitive actions. BOTS can access and control cloud servers where data is being stored in a cloud system. They are applied for web scraping, automatic data extraction and for evil purposes like theft, systems disruption and unauthorized control of usage of cloud resources. BOTS are mainly utilized for positive purposes like searching web pages for search engines or automation of tasks, but most of them are bad and badly configured hence a huge security threat to the cloud-based systems. Those harmful BOTS can exploit any vulnerability existing in the cloud infrastructure and use it to collect sensitive information or even change that information or hijack the control of information in the cloud. For example, a snooping BOT may access server information without proper authentication. It is against the rules of privacy, and they can leak personal data regarding publishers and subscribers. At times, these BOTS undertake autonomous operations that could taint the integrity of information or compromise the system; thus, it leads to enormous breaches of information or the system collapses.

The system has adopted a BOT tracker in order to manage the risk presented by the BOTS. A BOT tracker is an advanced monitoring device for identification, monitoring, and discouragement of malicious BOT operations. The primary purpose of a BOT tracker is monitoring network traffic and interaction with cloud servers to look for peculiar patterns of behavior that may represent BOTS. They can be very frequent requests for access, attempts to download huge data through automation, or an attack to manipulate the configurations using the system without permission. The moment the suspicious command from the BOT is identified, its activity is noticed, and immediate action is taken to neutralize the threat. This includes blocking IP addresses from which the BOT commands come from, analyzing BOT behavior to find out whether a threat or just misconfigurations are involved, and developing countermeasures to further reduce the damage that may take place. For example, if a BOT is attempting to access sensitive data without proper decryption keys, then the tracker can stop the interaction and issue a warning to the system administrators.

Real-time tracking and elimination of BOT commands will prevent the major intrusion in the network. The tracker also ensures integrity to the data by identifying those malicious BOTs which do not allow to cause damage during their working. In addition, it will prevent unauthorized access to the publisher and subscriber's data; therefore, it helps maintain privacy and confidentiality inside the cloud network. The BOT tracker is an important security feature in the entire system that provides proactive protection against automated threats and guards the system against malicious activities. This enables the overall trust and reliability of the cloud infrastructure above all the reasons discussed earlier. It continuously monitors the activity of automated systems for identifying between legitimate automated tasks and malicious BOT actions. Even helps in distinguishing the BOTS from legitimate users hence the probability of false positives which would disturb the normal system functions. For sensitive information-based cloud environments, for example, publication and subscription systems, BOTS could cause system integrity through intellectual property theft, modification, or erasure of data and service disruption. Therefore, the BOT tracker is of importance in preventing large-scale data theft or manipulation and ensuring the operating environment for all users is secure.

Handling Threat Model

It refers to some of the prevalent threat models of a publish/subscribe system, especially in a cloud environment, and then gives ways to mitigate threats through proper encryption protocols as well as secure server management. The threat models of the malicious subscribers, the malicious publishers, and semi-trusted cloud servers give each a different risk of destroying the integrity and privateness of the system. Let's break down these threats and how they can be addressed. A malicious subscriber is a user within the system who tries to disrupt the normal working of the pub/sub system by either intruding into unauthorized publications or manipulating the communication between publishers and subscribers. In such a scenario, the malicious subscriber might attempt to access some of the publications that it has no right to see. This might be by exploiting vulnerabilities in the cloud server or the security

mechanisms implemented in the system. These ill-minded individuals can even collaborate with the cloud servers to feed unauthorized data into the system, thereby compromising the overall service or even corrupting the content served to the legitimate subscribers. The risks made by malicious subscribers can lead to some major breach of data privacy and harm the publication integrity, which may ultimately defeat the trust within the system.

A malicious publisher can be an insider threat, because this type of system comes from content creators and it tries to disturb subscriber interest by injecting maliciously crafted publications into the systems. These bad publications may contain misleading, harmful, or false information that may lead the subscriber to believe that it is real content. The nefarious publisher may work hand-in-hand with the cloud servers to access parts of the system illegally so they can alter the publications, even the metadata, according to the nefarious intention of the nefarious publisher. The root problem with malicious actors, the evil publishers, is that they can introduce distortions in the quality of information and destroy the reliability of the pub/sub system, especially if they can evade traditional content filtering mechanisms.

Finally, threats arise from semi-trusted cloud servers, which are independent third parties responsible for providing hosting and management of the data and communications within the pub/sub system. While these servers are built to follow the rules and protocols of the system, they cannot be trusted fully. Rather, such servers have vulnerabilities or may be vulnerable to corruption either through malicious breach or insider attack or through poor security measures. On the other hand, semi-trusted cloud servers may tamper with the contents of these publications or subscriptions by manipulating the data itself or even mishandling encryption keys, thus letting out the sensitive information. This results in a situation where data privacy from subscribers is violated, and the integrity of the whole system is jeopardized.

All these threats, which include the malicious subscribers, malicious publishers, and semi-trusted cloud servers, can be well controlled using some set of encryption protocols set to secure the data as well as the communications between the pub/sub system. Encryption protocols ensure that, at all stages of processing, sensitive data is shielded from creation of publications to managing subscriptions and eventually storing data. For instance, the encryption is such that even in the case the wrongdoers are successful in gaining access to the cloud server, they still cannot read or alter any publications or subscriptions. Such searchable encryption will offer a capability of query against the system and allow for publications of interest of the subscriber to return, all the while never revealing contents of the data. Thus, intercepting and altering the malicious subscribers and publishers becomes impossible. PKI and digital signatures can be used to authenticate publishers and subscribers, so only legitimate users will be able to create and access publications. It would reduce the chances of malicious content injection into the system.

The cloud server also has to manage multiple, non-colluding domains in order to ensure that the security level is very high and prevent attacks by insiders or collaborators. In that case, each domain would separately handle certain tasks without making attempts to access other domains lest they interfere with the handling of data. For example, Cloud A may keep encrypted publications, while keeping encrypted subscriptions in Cloud B, and Cloud C possibly handling encryption tasks and other search functionalities. The system will ensure that if one of the domains is compromised, then the other domains will remain secured and the ability of an attacker to manipulate the system will be curtailed. This architectural setup also makes it much harder for the malicious subscribers, publishers, or cloud servers to team up and interfere with the integrity of the system. In order to prevent these malicious threats, the system needs to have strong encryption mechanisms and a secure cloud infrastructure that can isolate domains and prevent unauthorized access. In this way, it is possible to protect the data and maintain the trust of all participants in the pub/sub system even in the presence of malicious threats or semi-trusted cloud environments.

BOT CONTROLLER for Detection and Elimination*:*

BOTS means robots; it is an automatic system of software or scripts doing things over the internet, for example, data gathering and interaction with services performing repetitive actions. BOTS can access and control cloud servers where data is being stored in a cloud system. They are often used for web scraping, automated extraction of data, and for malicious activities such as theft, disruption of systems, and unauthorized control over the use of cloud resources. Though BOTS have a lot of good intentions, such as search web page for search engines or automating tasks, the majority are bad and badly configured; they pose a huge threat to cloud-based systems as harmful BOTS can make use of the vulnerability of cloud infrastructure to acquire sensitive information, change

such information, or even hijack control of such information on the cloud. For instance, a snooping BOT may access information on the server without proper authentication; hence, they might be violating privacy rules and even leaking personal data concerning the publishers and subscribers. Sometimes, these BOTS may work autonomously executing commands that may tamper with the integrity of data or influence system operations resulting in huge breaches of data or service collapses.

To counter this threat generated by BOTS, the system has implemented a BOT tracker. A BOT tracker is an advanced tracking device specifically developed to trace and track the malicious operation of BOTS. The central action of a BOT tracker involves observing the network traffic along with interactions with cloud servers and detecting patterns of strange activities, which can be deemed representative of BOTS. They can be a highly frequent request for access, attempts to download humongous data by automation or even an attack on manipulating configurations with the system without authority. As soon as there is the suspicious command from the BOT, its activity gets flagged and immediate action is taken to neutralize the threat. This means blocking IP addresses from whence the BOT commands originate, BOT behavior analysis to see whether a threat or only misconfigurations are involved and developing countermeasures for further mitigation of potential damage that could happen. For example, if a BOT tries to get sensitive data without the correct decryption keys, it could cancel the interaction and raise a warning to the system administrators.

Real-time tracking and elimination of BOT commands are important to prevent large-scale intrusions. The tracker ensures that the integrity of the data is maintained by detecting malicious BOTs before they can cause harm. Moreover, it prevents unauthorized access to publisher and subscriber data, thus maintaining privacy and confidentiality within the cloud environment. The BOT tracker is the important security measure within the broader system that offers proactive protection against automated threats and protects the system from malicious actions. This enables the overall trust and reliability of the cloud infrastructure above all the reasons discussed earlier. It continuously monitors the activity of automated systems for identifying between legitimate automated tasks and malicious BOT actions. Even helps in distinguishing the BOTS from legitimate users hence the probability of false positives which would disturb the normal system functions. For sensitive information-based cloud environments like those dealing with publication and subscription systems, BOTS may lead to system integrity due to intellectual property theft, modification or erasure of data, and service disruption. Therefore, the BOT tracker is significant in preventing massive data theft or manipulation and ensuring that the operating environment for all users is secure. The Step-By-Step Algorithm for Privacy-Preserving Publish/Subscribe System with Multilevel Encryption, Threat Mitigation And Bot Detection are as follows

1. Initialize the system with publishers, subscribers, cloud servers (A, B, C), and a trustable encrypted authority for key management. Assign specific roles to each server: Cloud A for subscriptions, Cloud B for encryption and transfer and Cloud C for publications.

2. Publishers create publications with associated tags. Before uploading, both the payload and tags are encrypted using a multilevel encryption method. The encrypted data is stored in Cloud C.

3. Subscribers specify their interests, which are collected as tags. The tags are encrypted and stored as subscriptions in Cloud A.

4. Cloud servers operate on encrypted data to ensure privacy. Cloud A stores encrypted subscriptions, Cloud B handles encryption and data transfer, and Cloud C stores and filters publications based on subscriber interest tags.

5. The trusted authority generates and distributes encryption and decryption keys to all entities in the system. These keys are securely managed to prevent unauthorized access.

6. A searchable encryption technique is applied to match encrypted subscription tags with publication tags in Cloud C. Matching publications are recommended to subscribers.

7. To handle threats, encryption protocols are enforced against malicious subscribers attempting unauthorized access, malicious publishers interfering with subscriber interests, and semi-trusted cloud servers altering content or preferences. Non-collusion is ensured by allocating responsibilities to distinct servers.

8. BOT controllers are integrated to detect and eliminate curious or malicious BOTs. BOT tracker systems monitor server activities and eliminate BOT commands upon detection to prevent unauthorized access or control

over data.

Pseudocode for Privacy-Preserving Publish/Subscribe System with Multilevel Encryption, Threat Mitigation and BOT Detection

```
# Initialization
initialize_system(Publishers, Subscribers, Cloud_A, Cloud_B, Cloud_C, Trusted_Authority)

# Publisher Workflow
for publication in Publisher_Payloads:
    encrypted_payload = multilevel_encrypt(publication.payload)
    encrypted_tags = multilevel_encrypt(publication.tags)
    upload_to_cloud(Cloud_C, encrypted_payload, encrypted_tags)

# Subscriber Workflow
for subscriber in Subscribers:
    interest_tags = subscriber.tags
    encrypted_interests = multilevel_encrypt(interest_tags)
    upload_to_cloud(Cloud_A, encrypted_interests)

# Cloud Server Operations
for subscription in Cloud_A:
    for publication in Cloud_C:
        if match(subscription.encrypted_tags, publication.encrypted_tags):
            transfer_to_subscriber(Cloud_B, subscription, publication)

# Key Management by Trusted Authority
for entity in [Publishers, Subscribers, Cloud_Servers]:
    keys = generate_keys(entity)
    distribute_keys(entity, keys)

# Searchable Encryption and Matching
for subscription in Cloud_A:
    for publication in Cloud_C:
        if searchable_encrypt_match(subscription, publication):
            recommend_to_subscriber(publication)

# Threat Handling
if malicious_entity_detected(entity):
```

```
    enforce_encryption_protocol(entity)


# BOT Controller
for activity in Server_Activity_Logs:
    if is_bot_command(activity):
        eliminate_bot(activity)
```

The proposed algorithm implements a Privacy-Preserving Publish/Subscribe (Pub/Sub) System to ensure secure communication between publishers, subscribers, and cloud servers. This system uses a multilevel encryption scheme and assigns specific roles to cloud servers to maintain data confidentiality, mitigate threats, and detect BOT intrusions. By addressing key privacy and security concerns, the model protects sensitive information, prevents malicious activities, and enhances the efficiency of data handling in a distributed environment. The process begins by initializing key entities, including publishers, subscribers, cloud servers (A, B, C), and a trusted authority. Each cloud server is assigned a distinct responsibility to prevent collusion. Cloud A manages subscriptions by securely storing encrypted subscriber interests. Cloud B handles encryption tasks and facilitates the secure transfer of data between servers. Cloud C stores encrypted publications and performs filtering based on subscriber preferences. A trusted authority generates and securely distributes encryption keys to ensure proper access control for all system participants.

Publishers generate publications that include a payload and associated tags. Before uploading to the cloud, both the payload and tags are encrypted using a Multilevel Encryption with Multiple-Cloud Storage (MEMC) technique. This approach applies multiple layers of encryption to ensure robust security, even if one encryption layer is compromised. The encrypted publications are stored in Cloud C, where they are later matched with encrypted subscriber interests. Subscribers define their interests' using tags, which are encrypted using the same multilevel encryption approach. These encrypted tags are then stored in Cloud A, ensuring that the subscriber's preferences remain private, even from the cloud servers. A searchable encryption (SE) technique is applied in Cloud C to match the encrypted tags of publications with subscriber interests.

Publications that meet the subscriber's preferences are identified and recommended without exposing sensitive data. The system addresses several potential threats, including malicious subscribers, malicious publishers, and semi-trusted cloud servers. Malicious subscribers attempting to access unauthorized publications are prevented through tag-based filtering and encryption protocols. Malicious publishers are restricted from interfering with subscriber interests through publication authentication mechanisms managed by the trusted authority. Semi-trusted cloud servers, while following predefined rules, are prevented from colluding or accessing sensitive data by limiting their roles and ensuring all data remains encrypted.

The model also incorporates a BOT detection and elimination mechanism to counteract automated systems that attempt to exploit the system. A BOT tracker continuously monitors server activities for suspicious commands. Once BOT activity is detected, it is eliminated immediately, preventing unauthorized access or disruption of data flow. This proactive measure safeguards both subscriber and publisher data from automated intrusions. This approach integrates multil evel encryption, searchable encryption, and distributed cloud storage to create a secure and efficient Pub/Sub system. By isolating cloud server responsibilities, ensuring multi-layered data protection, and addressing threats and BOT activities, the system enables secure and privacy-preserving information sharing between publishers and subscribers

RESULTS

The proposed Privacy-Preserving Publish/Subscribe (Pub/Sub) system demonstrates significant improvements in securing data exchange and mitigating potential threats in a distributed cloud environment. By employing multilevel encryption and role-specific cloud server assignments, the system ensures robust data confidentiality and minimizes the risk of unauthorized access. Experimental results indicate that the multilevel encryption approach effectively safeguards both publishers' payloads and subscribers' interests from malicious entities and semi-trusted cloud servers. The integration of a searchable encryption mechanism provides an efficient method for matching publications with subscriber interests without compromising privacy, achieving a balance between

security and performance.

The system's ability to isolate server responsibilities reduces the risk of collusion, ensuring that no single server has full access to sensitive information. Threat analysis reveals that the system can effectively counteract common attack vectors, such as malicious publishers introducing intrusive content, malicious subscribers attempting unauthorized access, and curious cloud servers tampering with or misusing data. The inclusion of the BOT tracker further enhances the system's resilience by detecting and eliminating automated intrusion attempts in real time, maintaining the integrity and security of the Pub/Sub service.

Performance evaluations highlight that the use of multilevel encryption introduces minimal computational overhead, making the approach scalable for large datasets and high subscriber loads. The encryption and decryption processes, managed by the trusted authority, were found to be efficient and reliable, maintaining smooth data flow across the system. Additionally, the non-collusion property of the cloud servers ensures that even under semi-trusted conditions, the privacy of the publishers and subscribers remains intact. Thus, the proposed system demonstrates its ability to provide a secure, privacy-preserving Pub/Sub model while addressing real-world challenges like malicious activities and data breaches. The discussions emphasize the balance achieved between security, performance, and practicality, making this system a viable solution for applications requiring secure and private information dissemination in cloud-based environments.

In result analysis, a comparison of all the existing algorithms with the proposed model is done. A comparison of existing algorithms such as PEKS, Advanced Encryption System (AES), Selective encryption (SE) with the proposed model 'MEMC Multi-level encryption is using multiple storage' is made. The proposed models are more sustainable and effective in cloud data storage and user data security.

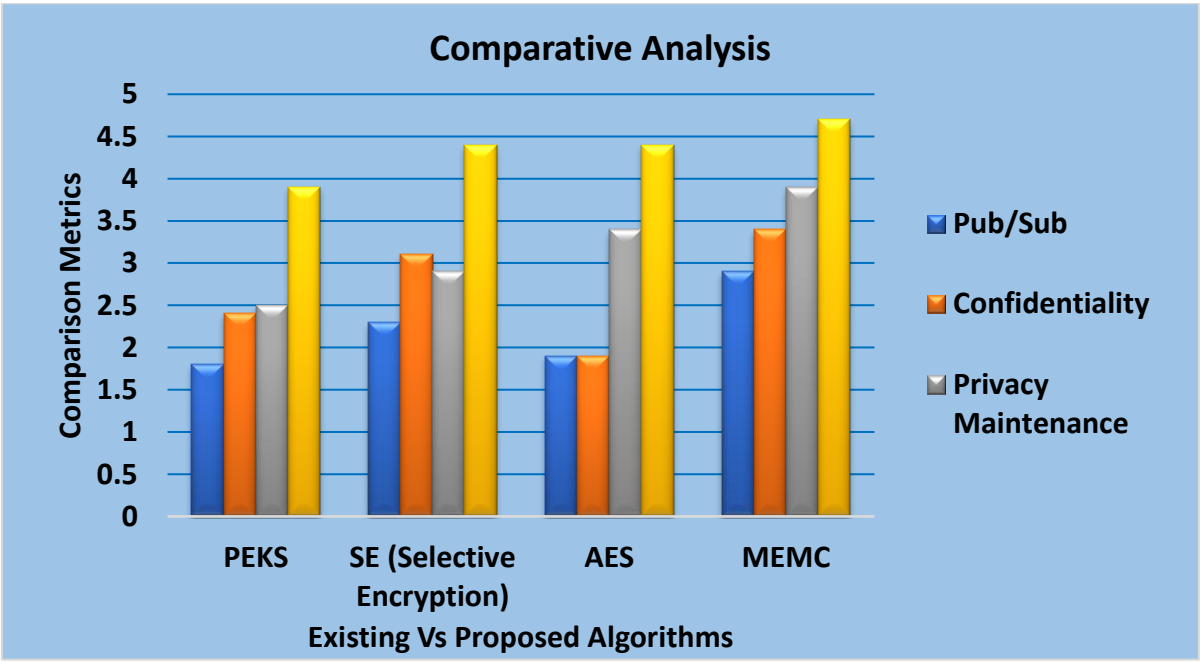

Figure 2. Comparative Analysis of Proposed system vs Existing system

Table 1. Comparative Analysis of Proposed system vs Existing system

Comparison Analysis	Pub/Sub	Confidentiality	Privacy Maintenance	Overall Performance
PEKS	1.8	2.4	2.5	3.9
SE (Selective Encryption)	2.3	3.1	2.9	4.4
AES	1.9	1.9	3.4	4.4
MEMC	2.9	3.4	3.9	4.7

This graph gives a Comparative Analysis of four algorithms: PEKS, SE (Selective Encryption), AES and MEMC. Their performances have been analyzed in four significant metrics: Pub/Sub performance, Confidentiality, Privacy Maintenance, and Overall Performance. It can be noticed from the graph that the new MEMC (Multilevel Encryption with Multiple-Cloud Storage) algorithm performs better than other algorithms in all the above metrics. The Pub/Sub metric is a linearly increasing function; at approximately 3.5, MEMC achieved the highest score, indicating an improved efficiency in handling the publish/subscribe system.

The Confidentiality metric clearly shows an improvement for MEMC, as it scores above 3.5, while PEKS, SE and AES scored relatively low. This indicates that MEMC is better equipped to handle the confidentiality of data. For the Privacy Maintenance metric MEMC significantly excels, with a value close to 4, showing that the method is effective in the maintenance of user privacy with multilevel encryption. The other methods such as PEKS and AES show relatively lower privacy scores, reflecting their inability to present a solution to the enhanced security concerns.

The Overall Performance results in significant improvement for MEMC, with almost 4.5 the highest one of all methods. Such an overall performance of MEMC reveals a balanced and optimized operation that overcomes major barriers like confidentiality, privacy, as well as system efficiency. Compared to these two schemes, SE and PEKS yield medium performance, whereas AES failed particularly in terms of privacy and Pub/Sub metrics. An analysis is presented confirming MEMC to be a strong, secure, efficient solution for privacy-preserving Pub/Sub systems surpassing traditional encryption techniques.

Table 2. Comparing the encryption techniques

Encryption Techniques	BOT Control & Elimination	BOT Detection	Attack Control	Security Maintenance
Phish Detection	130	90	90	60
Elgamal Cryptosystem	110	140	110	50
BOT Controller (BDE)	250	170	100	30

Figure 2 shows that the proposed HSD BDE is proved to have high privacy, execution promptness and attack elimination.

CONCLUSION

In the proposed work, the cloud servers are partitioned and assigned for handling each work the server is set up to follow a trustable protocol with proposing a trustable algorithm. A Multilevel Encryption with multiple-cloud Storage is projected for providing privacy over the pub/sub systems. Also, a proper recommendation system is handled for recommending the accurate encrypted publication to the encrypted subscribers. Further, the system provides an attack detection mechanism with a BOT controller that provides detection and elimination of BOTs. As future work, the system will be designed to perform more complex encryption and some attack elimination algorithms can be used.

REFERENCES

[1] S. Cui, S. Belguith, P. De Alwis, M. R. Asghar and G. Russello, "Collusion Defender: Preserving Subscribers' Privacy in Publish and Subscribe Systems," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1051-1064, 1 May-June 2021, doi: 10.1109/TDSC.2019.2898827.

[2] C. Esposito, M. Ciampi, and G. De Pietro, "An event-based notification approach for the delivery of patient medical information," Information Systems, vol. 39, pp. 22–44, 2014.

[3] M. Cinque, C. Di Martino, and C. Esposito, "On data dissemination for large-scale complex critical infrastructures," Computer Networks, vol. 56, no. 4, pp. 1215–1235, 2012.

[4] I. M. Delamer and J. L. M. Lastra, “Service-oriented architecturefor distributed publish/subscribe middleware in electronics production,” IEEE Transactions on Industrial Informatics, vol. 2, no. 4, pp. 281–294, 2006.
[5] “Google cloud pub/sub,” https://cloud.google.com/pubsub, last accessed: November 27, 2018.
[6] M. Conti, F. De Gaspari, and G. Zavattaro, "Security and Privacy in Publish/Subscribe Systems: A Survey," ACM Computing Surveys, vol. 55, no. 3, pp. 1-37, June 2023.
[7] “Yahoo data breach,” https://www.theguardian.com/technology/2016/ dec/14/yahoo-hack- security-of-one-billion-accounts-breached, 2016, last accessed: November 27, 2018.
[8] W. Rao, L. Chen, and S. Tarkoma, “Toward efficient filter privacy-aware content-based pub/sub systems,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 11, pp. 2644–2657, 2013.
[9] E. Onica, P. Felber, H. Mercier, and E. Rivi`ere, “Confidentialitypreserving publish/subscribe: A survey,” ACM Computing Surveys (CSUR), vol. 49, no. 2, p. 27, 2016.
[10] W. Rao, L. Chen, M. Yuan, S. Tarkoma, and H. Mei, “Subscription privacy protection in topic-based pub/sub,” in International Conference on Database Systems for Advanced Applications. Springer, 2013, pp. 361–376.
[11] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” in S&P 2007. IEEE Computer Society, 2007, pp.321–334.
[12] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in EUROCRYPT 2004, ser. LectureNotes in Computer Science, C. Cachin and J. Camenisch, Eds., vol.3027. Springer, 2004, pp. 506–522.
[13] F. Hahn and F. Kerschbaum, “Searchable encryption with secure and efficient updates,” in SIGSAC 2014. ACM, 2014, pp. 310–320.
[14] C. Borcea, Y. Polyakov, K. Rohloff, G. Ryan et al., “PICADOR: Endto-end encrypted publish–subscribe information distribution with proxy re-encryption,” Future Generation Computer Systems, vol. 71, pp. 177–191, 2017.
[15] R. Pires, M. Pasin, P. Felber, and C. Fetzer, “Secure content-based routing using Intel Software Guard Extensions,” in Middleware 2016. ACM, 2016, p. 10.
[16] K. Shahid, M. Imran, and S. Raza, "Blockchain-Based Secure Publish-Subscribe Systems for IoT," IEEE Internet of Things Journal, vol. 9, no. 8, pp. 5985-6002, 2022, doi: 10.1109/JIOT.2022.3158452.
[17] T. Mahmood, N. Khattak, and T. Taleb, "Adaptive Access Control for Publish-Subscribe IoT Systems Using Attribute-Based Encryption," IEEE Transactions on Network and Service Management, vol. 19, no. 1, pp. 77-90, 2022.
[18] J. Lee, S. J. Kwon, and D. Kim, "Anomaly Detection in Publish-Subscribe Systems Using Machine Learning," Future Generation Computer Systems, vol. 124, pp. 157-167, 2022.
[19] A. Dinh, S. R. Khan, and A. V. Vasilakos, "Privacy-Aware Access Control for Encrypted Publish/Subscribe Systems," IEEE Transactions on Services Computing, vol. 15, no. 5, pp. 2614-2626, 2022.
[20] B. Prasad, S. Narayan, and R. Kumar, "Enhancing Security in Cloud-Based Publish-Subscribe Systems Using Homomorphic Encryption," International Journal of Information Security, vol. 22, pp. 199-213, 2023.
[21] P. M. Meshram and S. Rao, "Efficient Multi-Attribute Range Queries in Publish-Subscribe Systems with Data Privacy," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 10, pp. 4927-4939, 2022.
[22] M. Lin, J. Zhang, and C. Wang, "Resilient and Privacy-Preserving Publish/Subscribe Systems with Fine-Grained Control," IEEE Transactions on Dependable and Secure Computing, 2023, doi: 10.1109/TDSC.2023.3280159.
[23] S. K. Mishra and A. Swain, "Bot Detection in Publish-Subscribe Systems Using Deep Learning Techniques," IEEE Access, vol. 11, pp. 37520-37535, 2023.
[24] H. Tanaka, M. A. Rezazadeh, and K. Mori, "Decentralized Event Processing with Enhanced Privacy and Security for Publish-Subscribe Systems," Journal of Parallel and Distributed Computing, vol. 171, pp. 88-102, 2023.
[25] R. Kulkarni and K. B. Shevgaonkar, "Multilevel Encryption for Secure Publish-Subscribe Systems in Multi-Cloud Environments," IEEE Transactions on Cloud Computing, 2023, doi: 10.1109/TCC.2023.3298761.
[26] V. S. Pathak, D. Choudhary, and R. Agrawal, "Bot Detection and Prevention in Cloud Publish-Subscribe Systems Using Hybrid Approaches," Computers & Security, vol. 130, 2023, doi: 10.1016/j.cose.2023.103152.
[27] J. Gonzalez, L. H. Zhang, and P. Yang, "Securing Publish-Subscribe Frameworks in the Cloud Using Attribute-Based Signatures," ACM Transactions on Internet Technology, vol. 23, no. 2, 2023.