Computer Communications 203 (2023) 262–275

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Privacy-preserving data dissemination scheme based on Searchable Encryption, publish–subscribe model, and edge computing

Ingrid Huso [a,b,*], Daniele Sparapano [a], Giuseppe Piro [a,b], Gennaro Boggia [a,b]

[a] Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy
[b] CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Italy

ARTICLE INFO

Keywords:
Searchable Encryption
Secure data dissemination
Internet of Things
Experimental evaluations

ABSTRACT

Attribute-based Searchable Encryption is emerging as a promising cryptographic technique supporting data protection, flexible access control, and keyword search over encrypted data. The current scientific literature already investigated its adoption in cloud-based services and additionally explored the usage of edge computing to implement some of the cryptographic tasks in scenarios with limited computational capabilities (such as Internet of Things). In the majority of the available solutions, however, the remote cloud is still responsible for data storage, keyword search over encrypted data, and delivery tasks. Indeed, the heavy computational load generated in scenarios with multiple data producers and data consumers (never studied yet) and large communication latencies can inevitably compromise the overall system performance. To bridge this gap, this work proposes a decentralized service architecture offering privacy-preserving data dissemination, by jointly leveraging attribute-based Searchable Encryption techniques, publish–subscribe communication model, and edge computing capabilities. Here, customized Edge Servers are deployed at the network edge to (i) collect subscription requests encoded via Searchable Encryption Trapdoors, (ii) receive data publications, encrypted via Attribute-based Searchable Encryption scheme, (iii) implement keyword search over encrypted data, and (iv) deliver encrypted data only to authorized requesters. Experimental tests explored the impact of network configuration and traffic load on both communication latency and energy consumption. Obtained results demonstrated the unique ability of the proposed solution to achieve shorter data delivery delays as well as less energy consumption with respect to cloud-based alternatives.

1. Introduction

Today, security and privacy are considered fundamental requirements for the communication infrastructures used in any application domain [1,2]. For this reason, many national and international regulations (like the General Data Protection Regulation in Europe [3]) impose the adoption of sophisticated mechanisms and tools ensuring confidentiality, data protection, access control, and other privacy-oriented security services. Moreover, to efficiently achieve this goal, some cryptographic algorithms, such as Attribute-based Encryption (ABE), have been conceived to natively enforce security directly on the data and prevent unauthorized entities (including *honest but curious clouds* [4]) accessing confidential information.

In this context, Searchable Encryption (SE) is emerging as a highly promising technique supporting data protection and keyword search over encrypted data [5]. Typically integrated into cloud-based applications, it assumes that [6]: (i) the data owner (also referred to as data producer) encrypts and uploads the data in the cloud, (ii) the authorized data users (also referred to as data consumer) issues a cryptographic keyword query to the cloud, and (iii) the cloud delivers specific encrypted data to the requester only in case of search matching.

Recently, the scientific literature investigated the usage of SE in Internet of Things (IoT) scenarios [7–9]. Moreover, given the high computational complexity of SE cryptographic operations, some contributions formulated lightweight techniques [10,11] or proposed to offload some security tasks to edge/fog nodes [8,12,13]. Nevertheless, despite these very valuable studies, available approaches still consider the remote cloud the only entity able to store data, manage keyword search over encrypted data, and deliver them to requesting users. Accordingly, the heavy computational load generated in scenarios with multiple data producers and data consumers (never studied yet) and larger end-to-end communication latencies inevitably compromise system performance.

Based on these premises, to achieve an important step forward in this direction, this work proposes a novel methodology offering a privacy-oriented data dissemination at the network edge. It is important to note that this contribution does not propose a new SE algorithm. Instead, it investigates the adoption of any SE algorithms within a

* Corresponding author at: Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy.
E-mail address: ingrid.huso@poliba.it (I. Huso).

https://doi.org/10.1016/j.comcom.2023.03.006
Received 30 September 2022; Received in revised form 10 January 2023; Accepted 7 March 2023
Available online 9 March 2023

List of Acronyms

5G	5th generation mobile network
ABAC	Attribute-Based Access Control
ABE	Attribute-Based Encryption
B5G	Beyond 5G
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
IoT	Internet of Things
JWT	JSON Web Tokens technology
KP-ABE	Key-Policy Attribute-Based Encryption
KPI	Key Performance Indicator
MEC	Multi-Access Edge Computing
MQTT	Message Queuing Telemetry Transport
NIST	National Institute of Standards and Technology
PBC	Pairing Based Cryptography
PKSE	Public-Key Searchable Encryption
POSE	Privacy-Oriented Search Engine
SE	Searchable Encryption
SSE	Symmetric Searchable Encryption
VM	Virtual Machine

novel service architecture handling the data dissemination process in a distributed manner and without the help of remote clouds. Differently from the current state of the art, this work provides the following main scientific contributions:

- First, it designs a novel service architecture supporting a scalable, efficient, and privacy-oriented data dissemination by combining attribute-based SE approaches, a publish–subscribe communication model, and edge computing capabilities. The reference architecture includes heterogeneous data producers and data consumers served by Beyond 5G (B5G) base stations and some Edge Servers deployed at the network edge. The former group of users generates and encrypts data via attribute-based SE and publish them to the closest Edge Server. While, the second group issues encrypted subscription requests (that are encrypted queries, namely *Trapdoors*) and share them with all the available Edge Servers. The keyword search over encrypted data is implemented at the network edge, e.g., by the Edge Servers, in a distributed manner. Indeed, once new data is published, the reference Edge Server delivers the encrypted data to the consumers that issued a valid Trapdoor.
- Second, it presents an implementation of the aforementioned service architecture modeling the most significant functions of an Edge Server and its interactions with data producers and other remote Edge Servers (including the receiving of data published by data producers on the Edge Server via a memoryless Poisson process, the execution on the Edge Server of the cryptographic operations, and the resulting data dissemination tasks). Here, communication latencies between logical nodes are enforced according to the values proposed in the literature.
- Third, it experimentally evaluates the performance of the proposed approach in realistic scenarios where coexist heterogeneous data producers and end-users. Note that the conducted study does not only investigate the computational complexity of SE operations as a function of security parameters (which represents the main objective of all the contributions available in the current scientific literature). Nevertheless, it explores the impact of network settings (i.e., number of Edge Servers) and loads (i.e., number of data consumers and various number of publications over time, modeled through Poisson-distributed rates) to these three Key Performance Indicators (KPIs): latencies associated with both cryptographic operations and the overall dissemination process, as well as the consumed energy. Obtained results demonstrate the unique ability of the proposed solution to achieve shorter delays and less energy consumption values compared to cloud-based alternatives.

The rest of the paper is organized as follows. Section 2 presents the background works and reviews the state-of-the-art on Searchable Encryption (SE) mechanisms. Section 3 illustrates the proposed methodology and provides technical details about the conceived communication protocol and procedures. Section 4 presents an experimental evaluation of the performance gains offered by the proposed approach with respect to the cloud-based one. Finally, Section 5 concludes the paper and draws future research activities.

2. Background concepts and literature review

This Section presents background concepts and reviews the state-of-the-art on Searchable Encryption.

2.1. Background concepts

Multi-Access Edge Computing (MEC). The hugely powerful performance requirements of 5G and B5G networks motivated the diffusion of the Multi-access Edge Computing (MEC) concept, which optimizes the spatial layout of network applications and services by utilizing pervasive computing, communication, and storage resources at the network edge [14,15]. According to ETSI-MEC specifications [16], each MEC host may integrate multiple MEC applications that, by taking advantage of strong tools and computational resources, are able to process (e.g., data mining and fusion) heterogeneous data produced by Internet of Things (IoT) devices as well as to offer cutting-edge and specialized services closer to the end-users.

Attribute-Based Access Control (ABAC). The National Institute of Standards and Technology (NIST) formulates a concrete solution for fine-grained authorization in dynamical IoT scenarios [17]. The ABAC methodology assumes that any resource is protected via dedicated access control policies, defined as a combination of access grants and properties. Indeed, to access a specific resource, an end-user must prove to be in possession of a subset of attributes satisfying the access control policy uniquely associated with the resource. Some noteworthy cryptographic algorithms directly include ABAC logic into the encryption and decryption procedures. Attribute-Based Encryption (ABE), Key-Policy Attribute-Based Encryption (KP-ABE) [18], and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [19] are among them. Indeed, these strategies may be used in IoT domain to provide data protection as well as flexible access control.

Searchable Encryption (SE). The growing number of IoT devices creates amounts of data to be collected and processed by means of computing and storage services (e.g., the Cloud). Indeed, to ensure privacy, the acquired data are encrypted before being stored in the public cloud [1]. However, despite the many advantages that cloud storage offers, protecting the privacy of sensitive data remains a difficult problem since the cloud servers are considered to be honest but curious [20]. This indicates that, although cloud service providers can be trusted for their services, they may also be interested in the data of their customers.

In this context, Searchable Encryption (SE) emerges as a preliminary turning point [5]. In cloud computing environments, SE offers a useful solution for issuing search queries on encrypted files based on specific keywords. Specifically, SE systems are constructed on a client/server architecture, in which the data owners and consumers serve as the clients during storage and retrieval, while the cloud acts as the server [20]. The data owner is responsible for outsourcing a collection of data and a list of keywords in an encrypted form [20]. The data user

Table 1
Review of related works.

Works	Multi-user	Multi-owner	Edge/Fog Computing	Privacy-preserving	Searchable encryption at the network edge	Data dissemination	Cryptographic security proof	Protocol security proof	Publish–subscribe model
[4]	✓			✓			✓		
[11]				✓			✓		
[12]			✓	✓			✓		
[13]			✓				✓		
[21]				✓			✓		
[22]				✓			✓		
[23]	✓			✓			✓		
[24]	✓			✓			✓		
[25]		✓		✓			✓		
[26]	✓			✓			✓		
[27]		✓		✓			✓		
[28]		✓		✓			✓		
[29]			✓	✓			✓		
[30]			✓	✓			✓		
[31]			✓	✓			✓		
[32]			✓	✓			✓		
[33]			✓	✓			✓		
[34]			✓	✓			✓		
[35]			✓	✓			✓		
[36]			✓	✓			✓		
[37]			✓	✓			✓		
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓

is authorized to retrieve data from the cloud by sending encrypted queries, namely Trapdoors, to the Cloud. The cloud server saves the documents submitted by the data owner and also handles search tasks: when a data user submits a Trapdoor, the cloud searches across the encrypted keywords and returns to the data user the documents that contain that specific keyword [20]. At the end, the data user can decrypt the received data [20].

2.2. Literature review

Recently, several research works in the field of IoT [8,9,38] emphasizes how the proliferation of smart devices results in huge amounts of data being exchanged through the network, reducing its security and raising the need to improve privacy and data protection. This underlines the significance of introducing new approaches capable of improving data transmission security.

In this context, the majority of existing solutions, present in the scientific literature, rely on cloud-based approaches where data is disseminated via remote clouds [10,11,39]. Herein, however, the server needs to be aware of the data sources, service type, end-users, and other relevant information in order to correctly transmit data to authorized end-users.

Thus, Searchable Encryption (SE) appears as a key enabling technology for providing privacy preservation, [5]. In cloud computing contexts, SE represents a valuable option for performing search queries on encrypted files based on specific keywords [1]. The work provided in [40] constitutes the first Symmetric Searchable Encryption (SSE) system in which the searchable ciphertexts are built using the symmetric key encryption approach, and users are able to create trapdoors using the shared key. Later, the contribution in [41] combines keyword searches with public key encryption methods, enabling users to safely recover the requested files over encrypted data using user-defined keywords. By utilizing both public and private keys, the Public-Key Searchable Encryption (PKSE) allows data owners and users to encrypt data with their public keys and create trapdoors with their secret keys. After then, a variety of PKSE scheme with different capabilities are presented in the scientific literature, including single keyword searches [42], fuzzy [43,44] and ranked keyword searches [45] as well as verified ones [46]. Since the aforementioned SE solutions are not feasible for bringing fine-grained search capabilities to end-users, recent studies [21–24,47–49] have recently looked at the integration of ABE schemes and SE techniques, significantly boosting ABE adaptability in a dynamic IoT environment. Herein, to guarantee high efficiency of encrypted data retrieval from cloud servers, the works in [25,26] introduce to a multi-user SE scheme, while, recent several works [27, 28], and [29] propose multi-owner scenarios. In IoT scenarios, new lightweight SE techniques are offered in edge and fog computing contexts as potential options for bringing data storage and processing capabilities closer to IoT devices [1]. Indeed, the studies in [30–32], and [33] describe four distinct SE schemes in fog-based IoT scenarios, where fog nodes help the cloud with searching and forwarding while also partially decrypting the documents that are retrieved to lessen the computational load on IoT devices. Moreover, [34,35] envision dynamical SE schemes with a multi-keyword search for smart grids in a cloud–edge architecture and edge computing respectively, where the search algorithm runs in collaboration between the edge nodes and the cloud server. While, recent works (such as [13,36,37,50], and [51]) offer distributed SE methods in different IoT domains, proving the possibility of fog and edge nodes to reduce the computational workload with regard to the cloud environment. Nevertheless, they equip fog or edge nodes with cryptographic capabilities to partly decode and encrypt searched queries.

2.3. Problem description

The review of related publications reveals that the present scientific literature focuses on the cryptographic features of the SE schemes, with the main purpose of finding and retrieving specific encrypted files. Furthermore, from the summary reported in Table 1 it is possible to conclude that:

- Available studies investigate the computational complexity of SE operations as a function of security parameters (i.e., number of attributes forming the access policy) and the number of files (i.e., the encrypted data) to be processed. Thus, none of them evaluate SE in realistic scenarios where coexist heterogeneous data producers and end-users.
- Most of the existing works leverage a cloud-based approach, where search and dissemination tasks are directly implemented by the remote cloud. In recent works, computational capabilities at the network edge have been used to implement encryption and decryption operations, thus limiting the complexity expected for constrained devices. However, the chance of performing SE operations directly at the edge of the network has not yet been investigated.

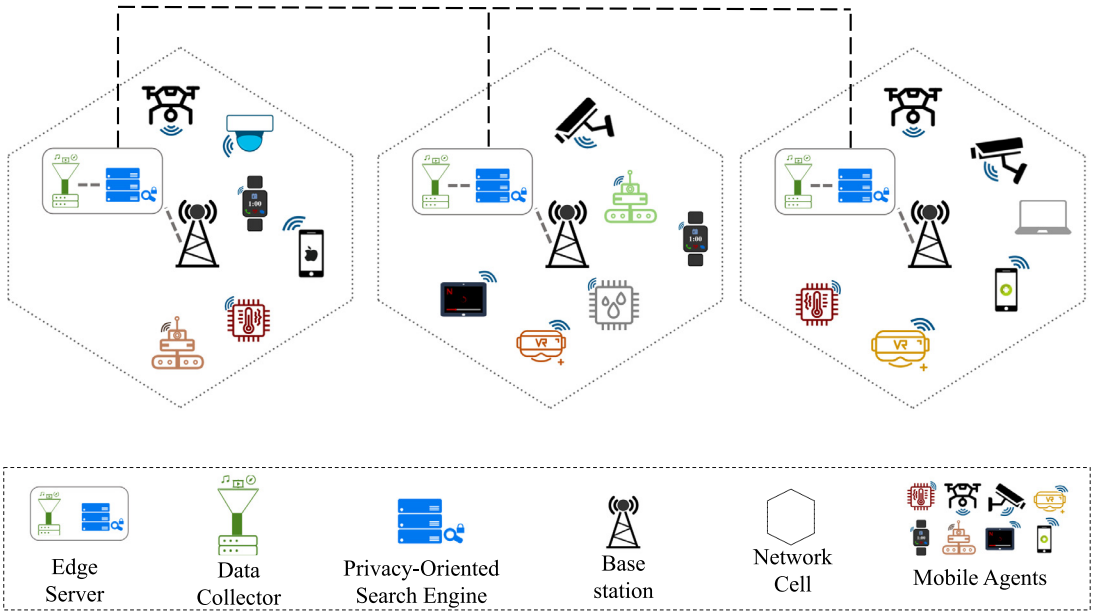


Fig. 1. The reference distributed service architecture.

- No contributions envisage the opportunity of sharing and disseminating data through the network, and in particular at the network edge, in a distributed, efficient, and privacy-preserved manner by using SE schemes and publish–subscribe model.

3. The conceived data dissemination scheme

Given the problem description presented in Section 2.3, this work presents a novel decentralized service architecture that provides privacy-preserving data distribution by combining Attribute-Based Searchable Encryption algorithms, publish–subscribe communication paradigm, and edge computing capabilities.

The reference architecture, shown in Fig. 1, is a typical B5G network, where base stations offer wireless connectivity to mobile agents and the edge network hosts computing platforms.

Here, mobile agents are divided into two specific groups: data consumers and data producers, properly distributed among different cells. Their interaction follows the publish–subscribe model. Data consumers issue their service requests by generating SE *Trapdoors* and share them across all the available servers, placed at the network edge. While, data producers encrypt data (e.g., AR/VR video stream, temperature or humidity data, and healthcare information from wearable sensors) via an Attribute-Based SE algorithm and publish them at the edge of the network.

Customized Edge Servers, deployed at the network edge, interact with each other and with data producers and data consumers to implement the following tasks: (i) collect subscription requests into a Trapdoor table, (ii) receive encrypted data published by data producers (iii) implement keyword search over encrypted data through SE cryptography, and (iv) deliver encrypted data only to authorized consumers. To this end, each Edge Server embraces a *Data Collector* and a *Privacy-Oriented Search Engine (POSE)*. These entities can be seen as two separate MEC applications running into the same MEC host, deployed close to the B5G base station, as depicted in Fig. 1. For sake of completeness, it is important to remark that the Data Collector entity is in charge of receiving the encrypted data published by data producers, while the POSE entity runs all the other main functionalities. Moreover, to guarantee a privacy-oriented approach each Edge Server is unable to retrieve any information from the requests since Trapdoors hide required service keywords using cryptographic schemes.

Differently from conventional and cloud-based architectures, the proposed service architecture implements SE operations at the network edge in a distributed way. The term “decentralized” refers to the possibility of distributing search operations at the network edge rather than assigning all to one single remote entity (i.e., in the cloud).

In summary, the following benefits are achieved:

- each Edge Server, and specifically each POSE entity, can implement keyword search operations only on encrypted data generated by the mobile agents served by a specific base station. This ensures a reduction of the overall computational burn with respect to cloud-based approaches and guarantees high levels of scalability in distributed environments;
- data dissemination managed at the network edge reduces communication latencies with respect to cloud-based approaches;
- the POSE entity implements the aforementioned keyword search over encrypted data by considering the list of subscriptions stored within the Trapdoor table. By storing in the Trapdoor table the details about the location of data consumers, it can be possible to deliver only once the encrypted data towards the base station serving consumers interested in the same data. This aspect ensures bandwidth and energy reduction.

3.1. Data dissemination workflow

This section describes both the search and data dissemination procedures by giving technical specifics concerning general security measures to be conducted.

It is noteworthy to mention that this contribution does not present a new SE method, but it rather tries to integrate one of the strategies currently available in the scientific literature to enable faster and privacy-oriented data dissemination at the network edge. As a result, any SE technique may be included into the entire data distribution procedure detailed here. Furthermore, to provide concrete examples, the technical details related to the SE algorithms presented in [7,36] and their integration in the proposed data dissemination scheme are reported in Appendices A and B, respectively.

The conceived data transmission procedure is organized into five different stages, as shown in Fig. 2 and explained below.

For sake of clarity, it is important to remark that the interaction between the trusted authority, data producers and consumers, and

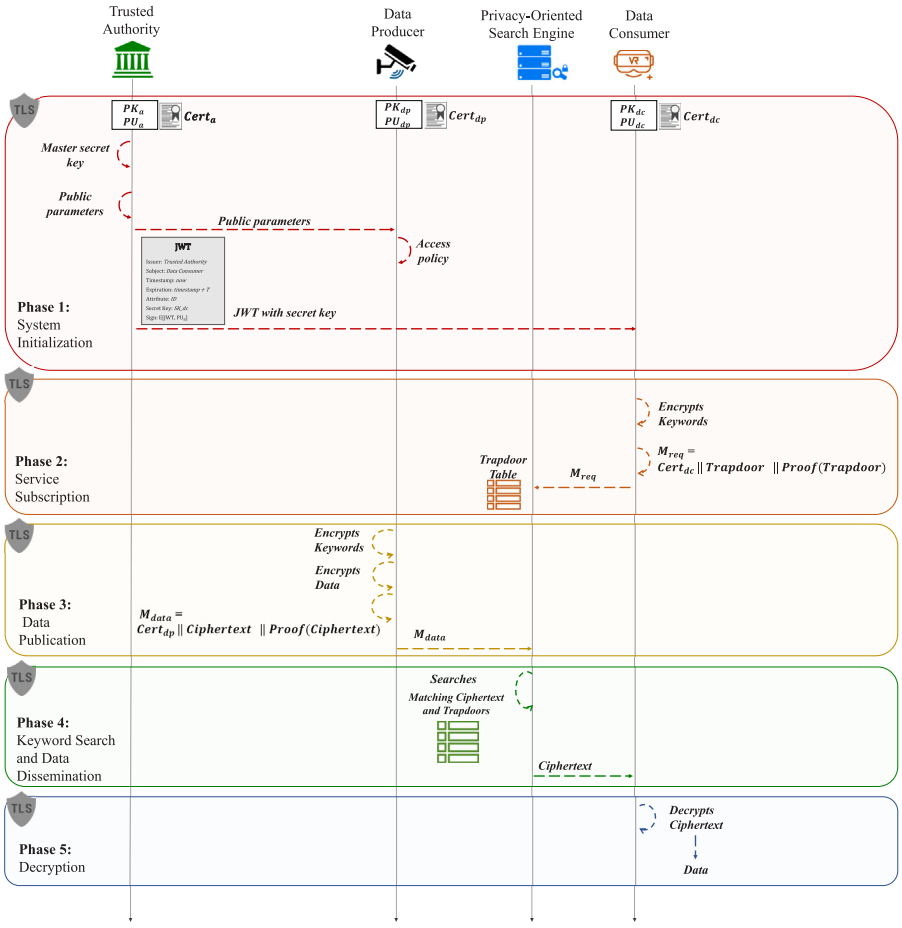


Fig. 2. General data dissemination scheme.

the POSE entity is protected via the Transport Layer Security (TLS) protocol.

Phase 1: system initialization. Phase 1 includes three different steps: initialization of public key cryptography, setup of the access control policies, and attributes processing. This phase involves a Trusted Authority which is responsible for enforcing system security. It is a completely trusted third party in charge of system configuration by addressing system security setups, key material creation, attribute management, and policy enforcement. Firstly, during this first phase, a private–public key pair is given to each entity in the network. Public keys are stored in trusted X.509 certificates. This will help achieving peer authentication within the following phases. Secondly, depending on the encryption algorithm, the Authority generates the master secret key and public parameters. The master secret key is kept secret by the Authority and it is used to create secret keys for data producers and consumers. Instead, the public parameters are exposed by the Authority to all the data producers in the network that use it and their attributes to generate their access control policies. Thirdly, each data consumer needs to obtain its attributes and its secret key. Token-based standard data structures, are used to send secret keys and associated attributes to data consumers.

A token is commonly used in literature as a container for security-related information, allowing to transfer authentication/authorization information among different communication entities. Basically, the token is a straightforward method that successfully implements the decoupling between authentication and authorization processes [52]. The conceived solution employs of the JSON Web Tokens technology (JWT) [53]. In particular, a JWT connects any type of information chosen by the token author (i.e., claims) to the identity of the data consumer for which the token was produced. Moreover, all the information within the token are encrypted [54]. Few standardized claims are already present in a JWT, such as the issuer (i.e., token generator), subject (i.e., token receiver), timestamp, and expiration date [53]. While, the secret key and a human-readable string of the attribute set are added. Finally, the sign field is annexed to the end of the container to guarantee the integrity and validity of the token.

Thus, the Authority sends a JWT to each data consumer that will be in possession of a series of attributes and the related cryptographic material.

Phase 2: service subscription. This phase leads data consumers to generate and publish search Trapdoors. Specifically, when a data subscriber needs to retrieve specific data, it chooses a set of keywords, takes its attributes and its secret key, and calculates the Trapdoor by encrypting it with an attribute-based encryption. Then, it asks for its certificate to the Trusted Authority and it generates a trusted and authentic service request message, namely M_{req}, as follows:

$$M_{req} = Cert_{dc} \| Trapdoor \| Proof(Trapdoor), \quad (1)$$

where $Cert_{dc}$ is the data consumer certificate, $Trapdoor$ is the above-created Trapdoor, and $Proof(Trapdoor)$ is the digital signature of the Trapdoor created by using the private key associated with the public key stored in the certificate. Finally, as previously described, the data consumer subscribes the service request message to all the POSE entities in the network.

Phase 3: data publication. Herein, before publishing data on the closest POSE entity, a data producer selects specific keywords associated with the data flow and encrypts them with the attribute-based encryption algorithm. It takes as input the data, the keyword, the secret

key, and the access policy, while it gives as output the ciphertext. Additionally, the data producer retrieves its certificate from the Trusted Authority and creates a trusted and authentic message, namely M_{data}, as follows:

$$M_{data} = Cert_{dp} \parallel ciphertext \parallel Proof(ciphertext), \quad (2)$$

where $Cert_{dp}$ is the data producer certificate, $ciphertext$ is the published data (protected with the ABE cryptosystem), and $Proof(ciphertext)$ is the digital signature of the ciphertext obtained with the private key linked to the public key stored in the certificate. Subsequently, the message is published on the referenced POSE entity.

Phase 4: keyword search and data dissemination. This phase involves the POSE entity, which runs the search algorithm to verify whether the published encrypted data matches one or more subscriptions stored in the Trapdoor table. Herein, differently from the current scientific literature, the proposed methodology addresses a scenario with multiple data producers and data consumers. In detail, for each received data, the search procedure progressively handles each subscribed Trapdoor. Thus, the POSE entity verifies their equality through specific equations depending on the cryptographic algorithm. The equation's validity demonstrates that (i) the set of keywords in the ciphertext contains the keywords of a subscribed request, and (ii) the data consumer attributes match the data producer access policy. The search algorithm returns 0 in case of mismatching, or 1 otherwise.

However, subscriptions of different data consumers connected to the same base station may result in a match. In this context, to guarantee energy efficiency at the network edge, the POSE entity disseminates only once the encrypted data towards the base station by listing all matched data consumers.

For sake of clarity, search and data dissemination operations are defined in the flow chart in Fig. 3.

Phase 5: decryption. Finally, the last phase allows the data consumer to decrypt the received cyphertext with the decryption algorithm taking as input the secret key and retrieving the data.

3.2. Security proof

This section formulates a security proof for the proposed service architecture, by jointly considering cryptographic operations and communication protocols.

Regarding cryptographic operations, it is significant to note that many works in the scientific literature proposing SE algorithms analyze the cryptographic security of their conceived technique. For example, the two algorithms integrated and evaluated in the proposed service architecture, respectively presented in [7,36], have been proven to be robust against both Chosen Keyword Attack (CKA) and Chosen Plaintext Attack (CPA). As a result, our envisioned solution is secured by design against the aforementioned cryptographic threats.

Regarding the security analysis of the rest of the service architecture, the designed solution leverages well-known security building blocks (such as TLS, X.509 certificates, and ABE cryptosystem). These remain independently constructed, and their security has been previously established and formally described in the reference contributions listed below. Thus the security analysis of each used functionality is proved as follows:

- **Secure end-to-end communication.** The proposed distributed service architecture, by using TLS (i.e., TLS version 1.3) protocol, guarantees the establishment of a secure end-to-end channel communication between each involved entity of the network (e.g., Trusted Authority, Data Producer, Data consumer, and POSE entity). Specifically, it helps providing data confidentiality and mutual authentication. Moreover, this makes the communication network resilient against Man-In-The-Middle (MITM) attacks. TLS is a well-known and extremely widespread security protocol, thus, its security proof has already been explored in [55,56], and [57].

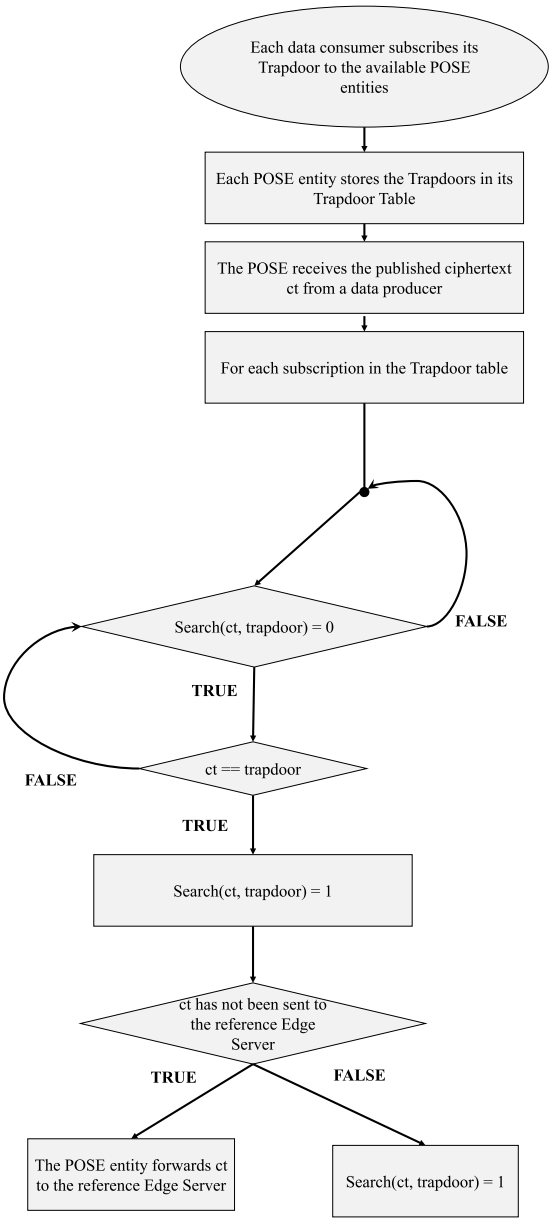


Fig. 3. Flowchart of the proposed search and data dissemination phase.

- **Peer and data authentication.** Independently from the usage of TLS, the service architecture presented in this work must ensure that submission requests and published data are generated by trusted entities. Therefore, data and peer authentication is achieved by using X.509 certificates and digital signature. In fact, the digital signature is a crucial aspect of the exchange of messages in the "service subscription" and "data publication" phases, as it serves to verify the authenticity and integrity of the message. This signature is generated using the private key associated with the public key included in the X.509 certificate, and is added to the message along with the mobile agent certificate and the Trapdoor or ciphertext (as shown in Eqs. (1) and (2)). Indeed, the robustness of the cryptographic technique used to generate the digital signature greatly affects the security of X.509 certificates. While classical algorithms such as ECDSA and RSA are commonly used and have been demonstrated to be secure in previous studies, [58,59] respectively. Newer quantum-resistant signature schemes such as CRYSTALS-Dilithium [60] and SPHINCS+ [61]

may also be used to ensure the security of the digital signature. Additionally, the security of the used standardized data structure determines the security of the cryptographic data associated with each data consumer (i.e., attributes and secret key). The security of the encrypted JWT token depends on the public-key cryptography algorithm used to create the digital sign. RSA and ECDSA techniques can be applied in this situation as well. Their security has been proven in [58,59], respectively. Thus, the peer and data authentication procedure helps the conceived methodology being resilient to collusion and replay attacks.

- **Access control.** The "data publication" and "decryption" phases make use of ABE cryptographic techniques (i.e., CP-ABE [19]) for guaranteeing data protection and flexible access control. In this context, the Trapdoor matches guarantee that data consumers asking for a specific resource prove to be in possession of a subset of attributes that satisfies the access control policy uniquely coupled with the resource and chosen by the data producers. This ensures that only authorized mobile agents access to the protected data.
- **Network traffic monitoring.** Intrusion Detection Systems (IDSs), as a unique network security approach, are an important defense solution [62]. Specifically, the IDS collects network traffic, security logs, and determines whether or not the network has been compromised by examining some indicators [62]. Thus, in line with recent scientific literature [63,64], and [65], the proposed service architecture can integrate IDSs at the network edge to ensure resistance to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Here, IDSs may be placed between the Edge Servers and the mobile agents, to filter and analyze encrypted traffic generated by data producers and received by data consumers. In this context, however, note that this contribution does not carefully investigate the behavior of IDS, which will be studied in future research activities.

4. Performance evaluation

To illustrate the significant potential of the proposed privacy-preserving data dissemination strategy in realistic scenarios, this Section investigates its performance through experimental tests.

4.1. The followed methodology

Differently from our preliminary numerical analysis presented in [66], the cryptographic operation expected for the two state-of-the-art SE algorithms proposed in [7,36] and presented in Appendices A and B, respectively, have been implemented through the Pairing Based Cryptography (PBC) library and executed within a proper experimental computing test environment modeling the conceived data dissemination process.

Conducted tests considers a network with a variable number of base stations (i.e., 4 and 8 cells). Herein, from 10 to 50 data consumers are uniformly distributed. Moreover, the study assumes to consider some data producers, uniformly attached to the base stations. Data producers may need to examine their data stream and publish new data because the network requirements are subject to change. To properly handle with this specification, the reviewed data publications related to the data producers are modeled via Poisson different rates (i.e., from 40 to 320 new publications/s) during a total time 180 s. Specifically, tests make use of the Message Queuing Telemetry Transport (MQTT) protocol which is based on a publish/subscribe communication mechanism. Publishers are clients who send messages, while subscribers are the ones who receive them. Their interaction is promoted by a central point (i.e., broker) that receives the messages from the publisher and delivers them to the subscribers [67]. Thus, the proposed methodology deploys MQTT to publish new data at different rates and to subscribe for receiving specific data. The impact of (i) the number of service

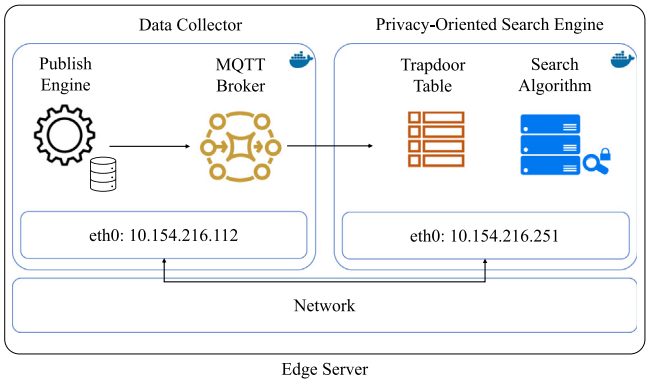


Fig. 4. Testbed setup.

subscriptions (equivalent to the number of Trapdoors generated by data consumer and stored within each single Trapdoor table), (ii) the average amount of data publications over time, and (iii) the number of Edge Servers involved into the privacy-preserving data dissemination strategy has been carefully studied by measuring three main Key Performance Indexes (KPIs). The first one is the *average search time*, defined as the average amount of time needed to complete the execution of the Searchable Encryption algorithm over all the service subscriptions stored within the Trapdoor table. The second one refers to the *average delivery delay*, computed as the average amount of time required to deliver published data to the subscriber. While the average search time highlights the computational impact of both the number of Trapdoors and network load on the implementation of the SE function, the average delivery delay extends the previous KPI by also reporting the impact of delivery delays of data across the distributed network. Finally, the third KPI regards the *energy consumption*. In particular, the consumed energy to run SE operations is examined as a function of both the number of subscriptions and the number of new publications changed. The analysis of these KPIs highlights the significant performance gain the proposed architecture achieves against a conventional cloud-based approach. Indeed, it demonstrates the benefits of distributing service subscriptions, collection of published data, SE tasks, and data delivery at the edge of the network rather than deploying a centralized and remote application available in the cloud.

4.2. System setup description

Experimental tests have been carried out on a workstation running the Ubuntu 22.04.1 LTS operating system, with an Intel Xeon Bronze 3106 @1.70 GHz processor, 96 GB of RAM and 180 W of consumed peak power.

Specifically, the workstation models the most important functionalities of an Edge Server and its interaction with respect to data publishers and other remote Edge Servers. Here, two lightweight networked containers are built by using Docker to model the Data Collector and the POSE entity (see Fig. 4).

The Data collector is in charge of collecting all data published by data producers served by a specific base station. Furthermore, it delivers the collected data to the POSE entity. As depicted in Fig. 5, these two tasks are executed by the Publish Engine entity and MQTT Broker, respectively. On the other hand, the POSE entity hosts the Trapdoor table and implements the SE algorithm. Moreover, the interaction between Data Collector and POSE entity is implemented through a client–server application, established by using the MQTT protocol version 3.1.1. In particular:

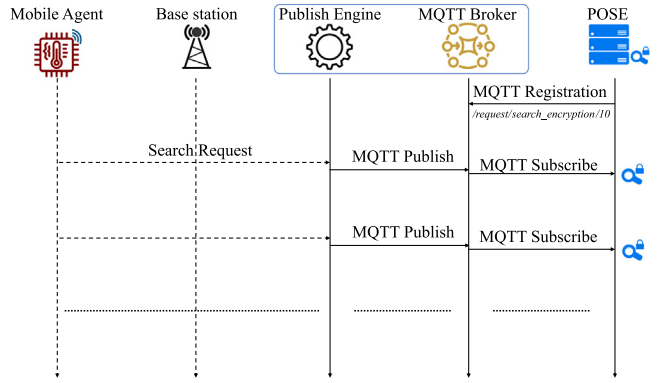


Fig. 5. System emulation scheme.

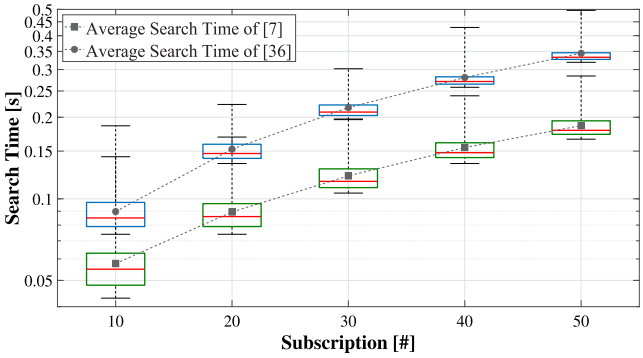


Fig. 6. Search time.

Table 2
Average communication latencies [68,69].

Network end-points	Average communication latency [s]
Radio interface	0.0036
Network edge	0.0052
Remote cloud	0.015

Table 3
Computational cost of cryptographic operations.

Cryptographic operation	Average execution time [s]	
	SE algorithm presented in [7]	SE algorithm presented in [36]
Encryption	0.02694	0.03264
Decryption	0.02568	0.03339
Search algorithm	0.02954	0.03233

- The Publish Engine uses a C++ script to emulate the reception of data published by IoT agents. Here, data are generated according to a memory-less Poisson process, where the average number of new data published over time in a single cell varies from 40 to 320 new publications/s.
- As soon as a new data is published, the Publish Engine in the Data Collector forwards MQTT Publish messages to the MQTT Broker (i.e., Eclipse Mosquitto message broker[1]), by using the Paho MQTT C library. Then, the MQTT Broker sends a MQTT Subscribe message to the POSE entity, previously registered to receive MQTT messages on a specific topic (e.g., `/request/search_encryption/trapdoor_number`).
- The POSE entity receives the MQTT Subscribe messages via the Flask API framework and runs the search algorithm by executing a binary file containing the cryptographic operations. Herein, to properly configure the scheme of the two attribute-based SE algorithms (i.e., [7,36]), which number of attribute is set to 5, Type A pairings are constructed on the curve $y^2 = x^3 + x$ over the field Z_P. Additionally, the dimensions of the $\mathbb{G}$ and $\mathbb{G}_T$ group elements are set to 1024 bits and to 160 bits for the Z_P ones.

In addition, the implemented experimental setup also emulates communication latencies on the various network segments to measure the average delivery delay. Table 2 shows latency values taken from the scientific literature [68,69].

[1] https://mosquitto.org/.

4.3. Analysis of cryptographic operations

This section evaluates the computational cost of cryptographic operations and measures statistical information on search algorithm execution time.

Firstly, by adopting the above described simulation setup and running 10^2 tests, the average execution time of encryption and decryption operations, as well as the single search accomplishment are calculated and reported in Table 3. It shows that the algorithm in [7] is less computationally intensive than the algorithm in [36].

Secondly, with respect to the conceived methodology presented in Section 3, the average search time is evaluated and defined as the amount of time needed to each POSE entity to execute sequentially the number of subscribed trapdoors. Fig. 6 depicts the statistical information of the search time execution run over 10^2 tests. For both the algorithms, the 25th, 50th, and 75th percentiles as well as the lowest value and maximum values, are reported. In addition, the respective average search time value is also shown. Thus, differently from the preliminary numerical results presented in our previous work [66], here the empirical results prove that the search execution time linearly increases with the number of subscriptions. Specifically, Fig. 6 demonstrates that the algorithm presented in [7] performs better than the one described in [36]. While, both algorithms highlight a double average search time in the execution of 10 sequentially Trapdoors with respect to the single execution. The results prove that when subscriptions pass from 10 to 50, the average search time triples in [7] and quadruples in [36].

4.4. Impact of the network load on the search time

This section analyzes the impact of network load on the search time execution, employing the simulation setup described in Section 4.2. Since POSE entities and cloud servers both have finite processing capability, herein it is proved that by gradually increasing the number of new published data, the number of queued search executions increases, exposing a longer average search time execution.

Table 4 displays the average search time execution, calculated as the average value of all single search execution performed on the new published data (generated with a specific Poisson distribution rate). Specifically, it shows the average search time execution of both considered algorithms (i.e., [7,36]) as a function of number of both submitted Trapdoors and network cells.

It emphasizes how moving search operations at the network edge would boost search operations on newly published data. Indeed, considering the algorithm in [7] with 10 subscriptions, by passing from 40 to 320 publications/s, the average search time execution within the proposed architecture increases of just few milliseconds (22 ms and

Table 4
Average search time execution.

Subscription no. [#]	Edge server no. [#]	New publication rate [publication/s]	Cloud-based average search time execution [s]		Proposed architecture average search time execution [s]	
			SE algorithm in [7]	SE algorithm in [36]	SE algorithm in [7]	SE algorithm in [36]
10	4	40	0.0770	0.1116	0.0685	0.1023
		80	0.0908	0.1369	0.0717	0.1049
		120	0.1228	0.2126	0.0746	0.1079
		160	0.1818	13.0792	0.0771	0.1116
		200	3.5898	37.2815	0.0815	0.1133
		240	21.5363	61.2140	0.0840	0.1158
		280	39.0132	84.1684	0.0855	0.1328
		320	43.1322	107.4534	0.0908	0.1369
	8	40	0.0771	0.1116	0.0673	0.1010
		80	0.0908	0.1370	0.0685	0.1023
		120	0.1227	0.2126	0.0694	0.1027
		160	0.1818	13.0792	0.0717	0.1049
		200	3.5898	37.2815	0.0718	0.1052
		240	21.5364	61.2140	0.07459	0.1079
		280	39.0132	84.1684	0.0757	0.1102
		320	43.1322	107.4534	0.0771	0.1116
40	4	40	0.1961	0.4105	0.1729	0.3063
		80	0.2775	53.9431	0.1785	0.3293
		120	33.2482	122.7198	0.1806	0.3742
		160	71.9170	190.0141	0.1961	0.4105
		200	110.0377	256.6541	0.1998	1.7370
		240	147.2419	281.9454	0.2178	19.3829
		280	187.1398	387.6209	0.2194	37.3034
		320	210.0170	445.3315	0.2775	53.9431
	8	40	0.1961	0.4105	0.1703	0.3063
		80	0.2775	53.9431	0.1729	0.3063
		120	33.2483	122.7199	0.1762	0.3211
		160	71.9170	190.0142	0.1785	0.3293
		200	110.0377	256.6541	0.1788	0.3438
		240	147.2419	281.9455	0.1806	0.3742
		280	187.1398	387.6210	0.1897	0.4020
		320	210.0170	445.3315	0.1961	0.4105

70 ms with 4 and 8 Edge Servers, respectively). While augments of 43 s with the Cloud-based approach. Similarly, the average search time execution for the suggested architecture rises of 53 s and 100 ms with 4 and 8 Edge Servers, respectively. While, it increases by 200 s in the Cloud-based scenario using the algorithm in [36] with 40 subscribers.

As a result, since the needed search time is not negligible, a technique that distributes tasks at the edge of the network might result in significant advantages.

4.5. Average delivery delay

By referring to the simulation setup specified in Section 4.2, this section compares the average delivery delay in the cloud-based and proposed scenario.

To properly evaluate the average delivery delay, the average communication latencies displayed in Table 2, the average time required to encrypt and decrypt data recorded in Table 3, and the average search time execution reported in Table 4 have been considered. On one hand, the average cloud-based delivery delay is evaluated as the sum of cryptographic operations and latencies due to reach the cloud server. On the other hand, following the proposed architecture, it is calculated as the sum of the following three contributions: cryptographic operations, latency in the radio interface, and latency experienced at the edge of the network.

Supposing to evaluate a scenario (e.g., monitoring and control applications) where the maximum acceptable delay is of 1 s, Fig. 7 and Fig. 8 depict the average delivery delay, respectively in a 4 and 8 cells network, as a function of new published data rates. Specifically, only below threshold values are displayed. In lines with the search time execution, the algorithm in [7] achieves shorter delivery delays than the algorithm in [36]. Moreover, results of both 10 and 40 subscriptions highlight how distributing search operations at the network edge allows obtaining a higher tolerable data publishing rate. Indeed, Fig. 7 shows that by passing from 10 to 40 subscriptions and using the algorithm in [7], the acceptable new published data rate for the proposed approach is not reduced, differently for the cloud-based one where it reduces by 60%. While, by exploiting the algorithm in [36], a maximum tolerated delivery delay is obtained reducing the new published data rate up to 10% for the proposed approach and 70% for the cloud-based one. Similarly, this happens in a 8 cells network depicted in Fig. 8.

By comparing the two figures, it is possible to understand that by increasing the number of cells and the new publication rates, the average delivery delay slightly varies in a scenario with 10 subscriptions for both the Cloud-based and proposed approaches. While it significantly differs by increasing the subscription number to 40. Indeed, on the one hand, the proposed service architecture allows to maintain a delivery delay lower than the threshold. On the other hand, the Cloud-based approach allows both algorithms suddenly reach the threshold value.

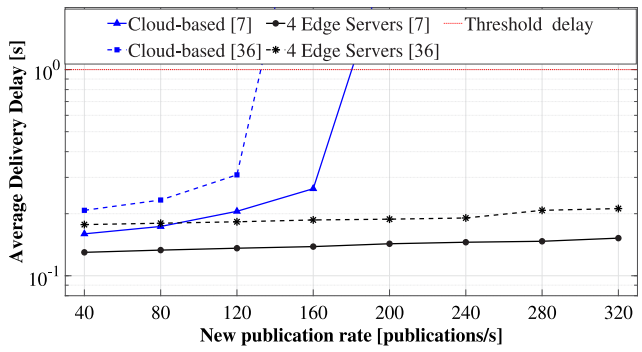
4.6. Energy consumption

The SE algorithms require a significant and not negligible amount of time and energy to run, both on Edge Servers and in remote clouds.

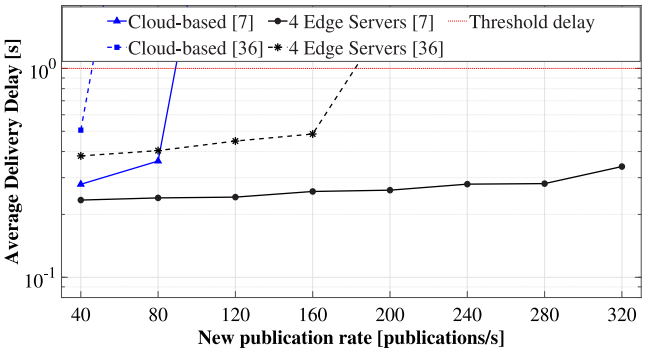
To measure the energy consumption due to the execution of *SE tasks*, conducted tests tracked the number of active and pending SE operations over time. Indeed, by dividing the time into small intervals ΔT (e.g., 1 ms each), the energy consumed during each interval is calculated using the percentage of peak power $P(n_{se})$, based on the number of active/pending SE operations (i.e., n_{se}), and each time slot ΔT. This allows evaluating the energy consumption E as:

$$E = \sum_{n_{\Delta T}} P(n_{se}) \cdot \Delta T, \tag{3}$$

where, $n_{\Delta T}$ is the total number of observed time slots. In this context, $n_{\Delta T}$ is the amount of time needed to run all the SE operations in a specific scenario.

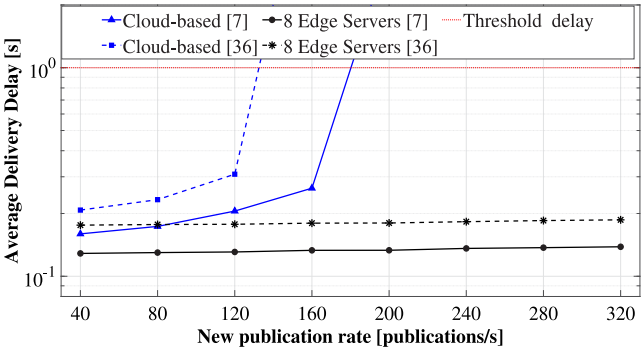


(a) Average Delivery Delay with 10 subscriptions

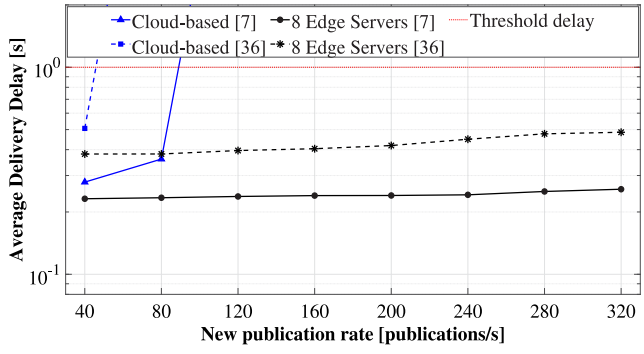


(b) Average Delivery Delay with 40 subscriptions

Fig. 7. Average delivery delay with 4 Edge Servers.

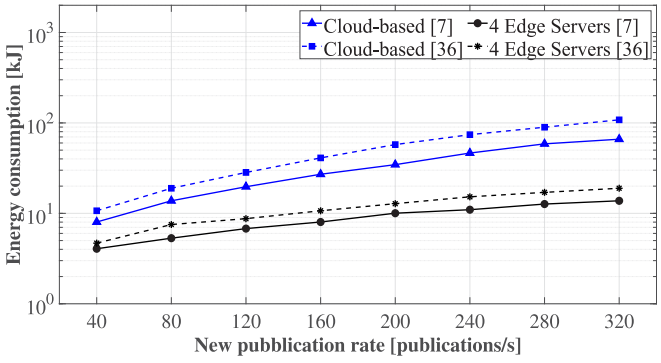


(a) Average Delivery Delay with 10 subscriptions

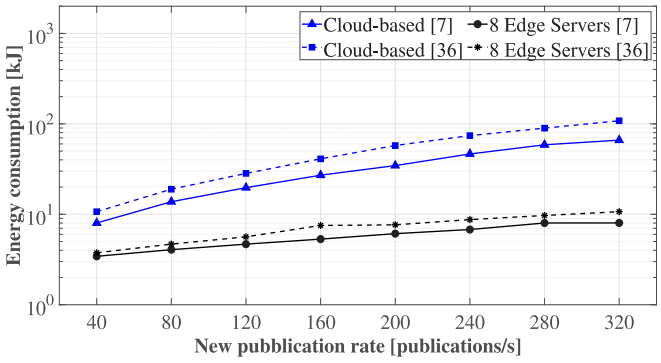


(b) Average Delivery Delay with 40 subscriptions

Fig. 8. Average delivery delay with 8 Edge Servers.



(a) 10 subscriptions



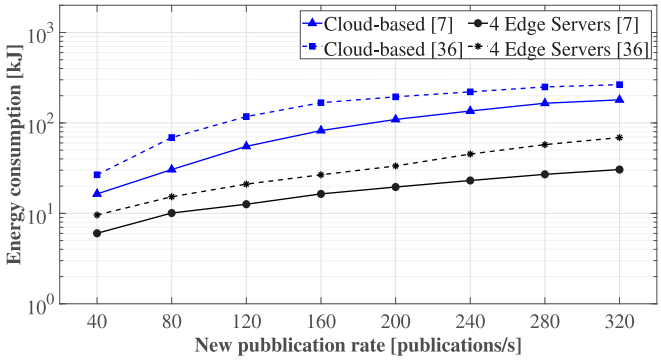
(b) 40 subscriptions

Fig. 9. SE energy consumption with 4 Edge Servers.

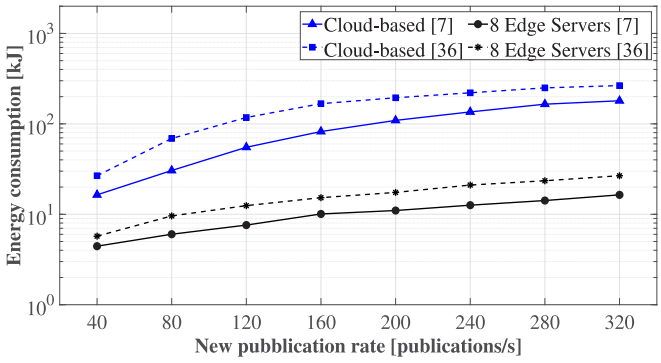
Fig. 9 and Fig. 10 illustrate the consumed energy in the function of the number of subscriptions, publication rates, and Edge Servers as well as the time needed to run the SE tasks. These figures confirm that the algorithm presented in [36] consumes more energy than the algorithm in [7]. It is due to the higher computational cost required by the algorithm in [36] to perform SE operations. Moreover, in a scenario with 4 Edge Servers and 10 subscriptions (Fig. 9(a)), the proposed solution permits keeping low energy values, ranging from 5 kJ to 20 kJ. Differently, the usage of the cloud-based method results in a higher energy consumption passing from 10 kJ to 100 kJ. In addition, in a scenario with 8 Edge Servers and 10 subscriptions (Fig. 10(a)), the consumed energy only increases by 8 kJ, passing from 40 to 320 publications/s within the proposed service architecture. Instead, with the cloud-based method, it increases by 90 kJ. The same pattern occurs in a network with 8 Edge Servers and 40 subscriptions (Fig. 10(b)). Here, the energy consumption values are lower than 20 kJ with the proposed approach, while it rises by a factor of ten, passing from 40 to 320 publications/s within the cloud-based one. By comparing the two figures, it is clear that the energy consumption resulting from the execution of SE operations in a scenario with 10 subscribers, for both the Cloud-based and proposed approach, marginally varies by increasing the number of cells and the new publishing rates. While it considerably differs by raising the number of subscriptions to 40, especially for the cloud-based methods.

5. Conclusions

This paper proposed a decentralized privacy-preserving data dissemination architecture based on Searchable Encryption, publish–subscribe communication model, and edge computing capabilities. The decentralized architecture embraces heterogeneous mobile agents attached to



(a) 10 subscriptions



(b) 40 subscriptions

Fig. 10. SE energy consumption with 8 Edge Servers.

Beyond 5G base stations and customized Edge Servers deployed at the network edge. In particular, Edge Servers collect subscription requests (i.e., Trapdoors), receive encrypted data publications, implement SE algorithms, and deliver encrypted data only to authorized requesters. The experimental results demonstrate that distributing Searchable Encryption operations at the network edge leads to significant advantages in terms of search time execution in overloaded networks, data delivery delays and energy consumption of SE tasks.

Future research activities will evaluate bandwidth consumption to better analyze the performance of the conceived approach. Simultaneously, they plan to formulate an optimized algorithm for distributing Edge Servers at the network edge based on traffic load, communication requirements, heterogeneous processing, and user dynamics. Lastly, an in depth-analysis of network traffic monitoring systems for the proposed service architecture will be studied.

CRediT authorship contribution statement

Ingrid Huso: Conceptualization, Methodology, Software, Writing, Validation. **Daniele Sparapano:** Software, Data curation, Writing – original draft. **Giuseppe Piro:** Conceptualization, Methodology, Review & editing, Supervision. **Gennaro Boggia:** Conceptualization, Methodology, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, with particular reference to the partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART", CUP: D93C22000910001) and to the national center on "Sustainable Mobility" (CN00000023 - program "MOST", CUP: D93C22000410001). It was also supported by the PRIN project no. 2017NS9FEY entitled "Realtime Control of 5G Wireless Net- works: Taming the Complexity of Future Transmission and Computation Challenges" funded by the Italian MUR, by "The house of emerging technologies of Matera (CTEMT)" project funded by the Italian MIMIT, and by the PON AGREED projects (ARS01 00254) fundend by the Italian MUR. The authors would like to acknowledge Marco Olivieri for the preliminary work done with the Pairing Based Cryptography library.

Appendix A. Cryptographic description of the searchable encryption algorithm presented in [7]

The technical details of the algorithm in [7] are described as follows.

Phase 1: system initialization. This attribute-based SE scheme considers two groups of order p, $\mathbb{G}$ and $\mathbb{G}_T$, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. At first, the trusted Authority randomly selects $\alpha, \gamma \in \mathbb{Z}_p$ and $g, h_1, h_2 \in \mathbb{G}$, and considers three hash functions $H_1, H_2, H_3 : \{0,1\} \rightarrow \{0,1\}^{log_p}$. Then, it generates the master secret key, that is M_k, and the public parameters, that are P_b, as in what follows:

$$\begin{cases} M_k = (\alpha, \gamma) \\ P_b = (g, g^{\alpha}, g^{\gamma}, h_1, h_2). \end{cases} \tag{A.1}$$

The master secret key, which is used to create users' secret keys, is kept private. The public parameters, instead, are published by the Authority. Moreover, by exploiting an AND-gate access structure based on n attributes and assuming that each attribute can assume different values, the Authority generates data consumers attributes set and data producers policies respectively denoted by: $\mathrm{X} = (x_1, x_2, ..., x_n)$ *and* $A = (a_1, a_2, ..., a_l)$. After receiving a set of attributes from the users, the Authority produces the secret key for that data consumer. Basically, a data consumer that joins the network sends its set of attributes to the Authority. Then, the Authority chooses a random $r \in \mathbb{Z}_p$ and implements the key generation algorithm:

$$\begin{cases} \rho_1 = (h_1 g^{-r})^{\frac{1}{\alpha - \sum_{i=1}^{n} H_1(x_i)}} \\ \rho_2 = (h_2 g^{-r})^{\frac{1}{\gamma - \sum_{i=1}^{n} H_1(x_i)}}. \end{cases}$$

Accordingly, the secret key of the data consumer, S_k, is computed as:

$$S_k = (r, \rho_1, \rho_2),$$

and shared with the reference data consumer.

Phase 2: service subscription. During this phase, the data consumer generates the search Trapdoor, that is t_{Φ}. Specifically, starting from its secret key S_k, the set of k keywords $\Phi = (\phi_1, \phi_2, \dots, \Phi)$ of its interest, and a random number $z_p \in \mathbb{Z}_p^*$, the Trapdoor is calculated as:

$$t_{\Phi} = (td_1, td_2, td_3), \tag{A.2}$$

where $td_1 = \rho_2^{z_p \cdot \sum_{i=1}^{k} H_2(\phi_i)}$, $td_2 = r \cdot z_p \cdot \sum_{i=1}^{k} H_2(\phi_i)$, and $td_3 = h_2^{z_p}$.

Then, the data consumer subscribes the Trapdoor to all the Edge Servers in the system.

Phase 3: data publication. Let M be the data to encrypt and to publish to the Edge Server. $\Psi = (\psi_1, \psi_2, ..., \psi_z)$ denotes the list of z keywords associated with that data. Moreover, $A = (a_1, a_2, ..., a_l)$ represents the list of attributes forming the access policy used to protect the data against unauthorized users. The encryption algorithms consider in input the public parameters P_b, the data M, the set of keywords Ψ,

and the access policy A. Indeed, by extracting a random $s \in \mathbb{Z}_p^*$, the ciphertext is obtained as:

$$ct = (C_1, C_2, C_3, v, C_4, C_5, C_6)\,, \tag{A.3}$$

where:

$$\begin{cases} C_1 = g^{\alpha s} \cdot g^{-s \cdot \sum_{i=1}^{l} H_1(a_i)} \\ C_2 = e(g,g)^s \\ C_3 = M \cdot e(g,h_1)^{-s} \\ v = H_3(C_1, C_2, C_3) \\ C_4 = g^{\gamma v} \cdot g^{-v \cdot \sum_{i=1}^{l} H_1(a_i)} \\ C_5 = e(g,g)^v \\ C_6 = g^{v \cdot \sum_{i=1}^{z} H_2(\psi_i)} \end{cases}$$

Phase 4: keyword search and data dissemination. This phase involves the POSE entity, which performs the search algorithm to determine whether the published encrypted data match one or more subscriptions stored in the Trapdoor table. In details, for each published data and for each stored subscription, the Edge Server verifies that the following equation holds:

$$e(C_4, td_1) \cdot C_5^{td_2} = e(C_6, td_3). \tag{A.4}$$

The validity of the equation proves that (i) the set of keywords Ψ in ct contains the keywords Φ retrieved from t_Φ and (ii) the set of attributes X belonging to the data consumer matches the access policy A used to protect the considered data. In case of matching, the search algorithm produces in output 0, otherwise it returns 1.

Phase 5: decryption. This phase allows the data consumer to decrypt the received cyphertext ct, by using its $sk = (r, \rho_1, \rho_2)$:

$$M = C_3 \cdot e(C_1, \rho_1) \cdot C_2^r. \tag{A.5}$$

Appendix B. Cryptographic description of the searchable encryption algorithm presented in [36]

The technical details of the algorithm in [36] are described as follows.

Phase 1: system initialization. This attribute-based SE scheme considers two groups of order p, $\mathbb{G}$ and $\mathbb{G}_T$, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Moreover, it takes g_0 and g_1 as G generators and $U = h_1, \ldots, h_u$ as the attribute set. At first, the trusted Authority randomly selects $\alpha, a \in \mathbb{Z}^*{}_p$ and $g_2, \hat{g_2}, h_1, h_2, \ldots, h_u \in \mathbb{G}$, and considers an hash function $H_1 : \{0,1\} \to \mathbb{Z}_p$. Then, it generates the master secret key, that is M_k, and the public parameters, that are P_b, as in what follows:

$$\begin{cases} M_k = (g_0{}^{\alpha}, a) \\ P_b = (H_1, g_0, g_0{}^{a}, g_2, \hat{g_2}, e(g_0,g_0)^{\alpha}, e(g_0,g_0)^{a}, U, \mathbb{G}, \mathbb{G}_T). \end{cases} \tag{B.1}$$

The master secret key, which is used to create users' secret keys, is kept private. The public parameters, instead, are published by the Authority.

Moreover, lets consider n attributes and assuming that each attribute can assume different values, the Authority generates data consumers attributes set and data producers policies respectively denoted by: $X = (x_1, x_2, ..., x_n)$ *and* $A = (a_1, a_2, ..., a_l)$.

After receiving a set of attributes from the users, the Authority produces the secret key for that data consumer. Basically, a data consumer that joins the network sends its set of attributes $X = (x_1, x_2, ..., x_n)$ to the Authority. Then, the Authority chooses a random $k, v, z \in \mathbb{Z}_p$ and implements the key generation algorithm:

$$\begin{cases} K = g_0{}^{\alpha} g_0{}^{a_k} \\ A = g_0{}^{v} \\ B = g_2{}^{a} \hat{g_2}{}^{v} \\ A_x = h_x^{v} \end{cases}$$

Accordingly, the secret key of the data consumer, S_k, is computed as:

$$S_k = (K, A, B, \{A_x\}),$$

and shared with the reference data consumer.

Phase 2: service subscription. During this phase, the data consumer generates the search Trapdoor, that is t_Φ. Specifically, starting from its secret key S_k, the keyword Φ of its interest, and a random number $u \in \mathbb{Z}_p^*$, the Trapdoor is calculated as:

$$t_\Phi = (td_1, td_2, td_3), \tag{B.2}$$

where $td_1 = B\hat{g_2}{}^{H_1(\Phi)}$, $td_2 = Ag_0{}^{uH_1(\Phi)}$, and $td_{3,x} = A_x * h_x^{uH_1(\Phi)}$.

Then, the data consumer subscribes the Trapdoor to all the Edge Servers in the system.

Phase 3: data publication. Let M be the data to encrypt and to publish to the Edge Server. $\Psi = (\psi_1, \psi_2, ..., \psi_z)$ denotes the list of z keywords associated with that data. Moreover, $A = (a_1, a_2, ..., a_l)$ represents the list of attributes forming the access policy used to protect the data against unauthorized users. The encryption algorithms consider in input the public parameters P_b, the data M, the keyword Ψ, and the access policy A. Indeed, by randomly extracting $d, f, \epsilon \in \mathbb{Z}_p^*$, the ciphertext is obtained as:

$$ct = (C_1, C_2, C_3, C_4, C_5, C_6, C_7)\,, \tag{B.3}$$

where:

$$\begin{cases} C_1 = F_d e(g_0,g_0)^{\alpha,d} \\ C_2 = g_0^d g_0^f \\ C_3 = g_1^d g^f \\ C_4 = g_0^d \\ C_5 = e(g_0^a, g_2)^{\epsilon} \\ C_6 = g_0^{\epsilon} \\ C_7 = g_0^{H_1(\Phi)\epsilon} \end{cases}$$

Phase 4: keyword search and data dissemination. This phase involves the POSE entity, which performs the search algorithm to determine whether the published encrypted data match one or more subscriptions stored in the Trapdoor table. In details, for each published data and for each stored subscription, the Edge Server verifies that the following equation holds:

$$\frac{e(td_1, C_6)}{C_5} = \prod_{x \in X} (e(C_{6x}, td_2) \cdot e(C_7, td_3)). \tag{B.4}$$

The validity of the equation proves that (i) the keyword Ψ in ct corresponds the keyword Φ retrieved from t_Φ and (ii) the set of attributes U belonging to the data consumer matches the access policy A used to protect the considered data. In case of matching, the search algorithm produces in output 0, otherwise it returns 1.

Phase 5: decryption. This phase allows the data consumer to decrypt the received cyphertext ct, by using its secret key:

$$M = \frac{e(g_0,g_0)^{\alpha d} \cdot e(g_0,g_0)^{kad}}{e(g_0^{\alpha+ak}, g_0^d)}. \tag{B.5}$$

References

[1] Y. Harbi, Z. Aliouat, A. Refoufi, S. Harous, Recent security trends in internet of things: A comprehensive survey, IEEE Access 9 (2021) 113292–113314, http://dx.doi.org/10.1109/ACCESS.2021.3103725.

[2] B. Ji, Y. Wang, K. Song, C. Li, H. Wen, V.G. Menon, S. Mumtaz, A survey of computational intelligence for 6G: Key technologies, applications and trends, IEEE Trans. Ind. Inform. 17 (10) (2021) 7145–7154, http://dx.doi.org/10.1109/TII.2021.3052531.

[3] European Parliament, Council of the European Union, The general data protection regulation, 2016.

[4] A. Ometov, O.L. Molua, M. Komarov, J. Nurmi, A survey of security in cloud, edge, and fog computing, Sensors 22 (3) (2022) http://dx.doi.org/10.3390/s22030927.

[5] T. Soo Fun, A. Samsudin, Recent technologies, security countermeasure and ongoing challenges of Industrial Internet of Things (IIoT): A survey, Sensors 21 (19) (2021).
[6] N. Andola, R. Gahlot, V.K. Yadav, S. Venkatesan, S. Verma, Searchable encryption on the cloud: a survey, J. Supercomput. (2022) 1–33.
[7] H. Wang, J. Ning, X. Huang, G. Wei, G.S. Poh, X. Liu, Secure fine-grained encrypted keyword search for E-healthcare cloud, IEEE Trans. Dependable Secure Comput. 18 (3) (2021) 1307–1319, http://dx.doi.org/10.1109/TDSC.2019.2916569.
[8] H. Wang, K. Fan, K. Zhang, Z. Wang, H. Li, Y. Yang, Encrypted data retrieval and sharing scheme in space–air–ground-integrated vehicular networks, IEEE Internet Things J. 9 (8) (2022) 5957–5970, http://dx.doi.org/10.1109/JIOT.2021.3062626.
[9] K. Wang, C.-M. Chen, M. Shojafar, Z. Tie, M. Alazab, S. Kumari, AFFIRM: Provably forward privacy for searchable encryption in cooperative intelligent transportation system, IEEE Trans. Intell. Transp. Syst. (2022) 1–12, http://dx.doi.org/10.1109/TITS.2022.3177899.
[10] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, M. Imran, Lightweight searchable encryption protocol for industrial internet of things, IEEE Trans. Ind. Inform. 17 (6) (2021) 4248–4259, http://dx.doi.org/10.1109/TII.2020.3014168.
[11] B. Chen, L. Wu, N. Kumar, K.-K.R. Choo, D. He, Lightweight searchable public-key encryption with forward privacy over IIoT outsourced data, IEEE Trans. Emerg. Top. Comput. 9 (4) (2021) 1753–1764, http://dx.doi.org/10.1109/TETC.2019.2921113.
[12] Y. Tao, P. Xu, H. Jin, Secure data sharing and search for cloud-edge-collaborative storage, IEEE Access 8 (2020) 15963–15972, http://dx.doi.org/10.1109/ACCESS.2019.2962600.
[13] Mamta, B.B. Gupta, M.D. Lytras, Fog-enabled secure and efficient fine-grained searchable data sharing and management scheme for IoT-based healthcare systems, IEEE Trans. Eng. Manage. (2022) 1–13, http://dx.doi.org/10.1109/TEM.2022.3143661.
[14] G. Gür, A. Kalla, C. de Alwis, Q.-V. Pham, K.-H. Ngo, M. Liyanage, P. Porambage, Integration of ICN and MEC in 5G and beyond networks: Mutual benefits, use cases, challenges, standardization, and future research, IEEE Open J. Commun. Soc. 3 (2022) 1382–1412, http://dx.doi.org/10.1109/OJCOMS.2022.3195125.
[15] K. Velasquez, D. Perez Abreu, M. Curado, E. Monteiro, Resource orchestration in 5G and beyond: Challenges and opportunities, Comput. Commun. 192 (2022) 311–315, http://dx.doi.org/10.1016/j.comcom.2022.06.019.
[16] Multi-access Edge Computing (MEC): Framework and Reference Architecture, ETSI GS MEC 003 v.2.1.1, 2019.
[17] Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162, 2014.
[18] M. Rasori, P. Perazzo, G. Dini, S. Yu, Indirect revocable KP-ABE with revocation undoing resistance, IEEE Trans. Serv. Comput. 15 (5) (2022) 2854–2868, http://dx.doi.org/10.1109/TSC.2021.3071859.
[19] J. Li, W. Yao, J. Han, Y. Zhang, J. Shen, User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage, IEEE Syst. J. 12 (2) (2018) 1767–1777, http://dx.doi.org/10.1109/JSYST.2017.2667679.
[20] U. Varri, S. Pasupuleti, K. Kadambari, A scoping review of searchable encryption schemes in cloud computing: taxonomy, methods, and recent developments, J. Supercomput. 76 (4) (2020) 3013–3042.
[21] K. Liang, W. Susilo, Searchable attribute-based mechanism with efficient data sharing for secure cloud storage, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1981–1992, http://dx.doi.org/10.1109/TIFS.2015.2442215.
[22] Y. Miao, X. Liu, R.H. Deng, H. Wu, H. Li, J. Li, D. Wu, Hybrid keyword-field search with efficient key management for industrial internet of things, IEEE Trans. Ind. Inform. 15 (6) (2019) 3206–3217, http://dx.doi.org/10.1109/TII.2018.2877146.
[23] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, H. Li, Practical attribute-based multi-keyword search scheme in mobile crowdsourcing, IEEE Internet Things J. 5 (4) (2018) 3008–3018, http://dx.doi.org/10.1109/JIOT.2017.2779124.
[24] Y. Bao, W. Qiu, P. Tang, X. Cheng, Efficient, revocable, and privacy-preserving fine-grained data sharing with keyword search for the cloud-assisted medical IoT system, IEEE J. Biomed. Health Inf. 26 (5) (2022) 2041–2051, http://dx.doi.org/10.1109/JBHI.2021.3100871.
[25] J. Cui, J. Lu, H. Zhong, Q. Zhang, C. Gu, L. Liu, Parallel key-insulated multiuser searchable encryption for industrial internet of things, IEEE Trans. Ind. Inform. 18 (7) (2022) 4875–4883, http://dx.doi.org/10.1109/TII.2021.3110193.
[26] J. Li, X. Wang, Q. Gan, F. Wang, MFPSE: Multi-user forward private searchable encryption with dynamic authorization in cloud computing, Comput. Commun. 191 (2022) 184–193.
[27] S. Abdelfattah, M. Baza, M.M.E.A. Mahmoud, M.M. Fouda, K.A. Abualsaud, M. Guizani, Multidata-owner searchable encryption scheme over medical cloud data with efficient access control, IEEE Syst. J. 16 (3) (2022) 5067–5078, http://dx.doi.org/10.1109/JSYST.2021.3123956.
[28] X. Tang, C. Guo, Y. Ren, C. Wang, K.-K.R. Choo, A global secure ranked multikeyword search based on the multiowner model for cloud-based systems, IEEE Syst. J. 16 (2) (2022) 1717–1728, http://dx.doi.org/10.1109/JSYST.2022.3157530.
[29] S. Gao, Y. Chen, J. Zhu, Z. Sui, R. Zhang, X. Ma, BPMS: Blockchain-based privacy-preserving multi-keyword search in multi-owner setting, IEEE Trans. Cloud Comput. (2022) 1–13, http://dx.doi.org/10.1109/TCC.2022.3196712.
[30] R. Zhou, X. Zhang, X. Wang, G. Yang, H. Wang, Y. Wu, Privacy-preserving data search with fine-grained dynamic search right management in fog-assisted Internet of Things, Inform. Sci. 491 (2019) 251–264.
[31] H. Li, T. Jing, A lightweight fine-grained searchable encryption scheme in fog-based healthcare IoT networks, Wirel. Commun. Mob. Comput. (2019).
[32] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, H. Li, Lightweight fine-grained search over encrypted data in fog computing, IEEE Trans. Serv. Comput. 12 (5) (2019) 772–785, http://dx.doi.org/10.1109/TSC.2018.2823309.
[33] Q. Zhang, G. Wang, W. Tang, K. Alinani, Q. Liu, X. Li, Efficient personalized search over encrypted data for mobile edge-assisted cloud storage, Comput. Commun. 176 (2021) 81–90.
[34] K. Fan, Q. Chen, R. Su, K. Zhang, H. Wang, H. Li, Y. Yang, MSIAP: A dynamic searchable encryption for privacy-protection on smart grid with cloud-edge-end, IEEE Trans. Cloud Comput. (2021) http://dx.doi.org/10.1109/TCC.2021.3134015.
[35] D. Wang, P. Wu, B. Li, H. Du, M. Luo, Multi-keyword searchable encryption for smart grid edge computing, Electr. Power Syst. Res. 212 (2022) 108223, http://dx.doi.org/10.1016/j.epsr.2022.108223.
[36] K. Gu, W. Zhang, X. Li, W. Jia, Self-verifiable attribute-based keyword search scheme for distributed data storage in fog computing with fast decryption, IEEE Trans. Netw. Serv. Manag. 19 (1) (2022) 271–288, http://dx.doi.org/10.1109/TNSM.2021.3123475.
[37] S. Niu, M. Song, L. Fang, F. Yu, S. Han, C. Wang, Keyword search over encrypted cloud data based on blockchain in smart medical applications, Comput. Commun. 192 (2022) 33–47, http://dx.doi.org/10.1016/j.comcom.2022.05.018.
[38] S.S. Chaeikar, A. Jolfaei, N. Mohammad, AI-enabled cryptographic key management model for secure communications in the internet of vehicles, IEEE Trans. Intell. Transp. Syst. (2022) http://dx.doi.org/10.1109/TITS.2022.3200250, 1–0.
[39] S. Li, S. Zhao, G. Min, L. Qi, G. Liu, Lightweight privacy-preserving scheme using homomorphic encryption in industrial internet of things, IEEE Internet Things J. (2021) http://dx.doi.org/10.1109/JIOT.2021.3066427.
[40] D.X. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000, 2000, pp. 44–55, http://dx.doi.org/10.1109/SECPRI.2000.848445.
[41] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in: C. Cachin, J.L. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004, Springer Berlin Heidelberg, 2004, pp. 506–522.
[42] I.R. Jeong, J.O. Kwon, D. Hong, D.H. Lee, Constructing PEKS schemes secure against keyword guessing attacks is possible? Comput. Commun. 32 (2) (2009) 394–396.
[43] R. Chen, Y. Mu, G. Yang, F. Guo, X. Wang, Dual-server public-key encryption with keyword search for secure cloud storage, IEEE Trans. Inf. Forensics Secur. 11 (4) (2016) 789–798, http://dx.doi.org/10.1109/TIFS.2015.2510822.
[44] P. Xu, H. Jin, Q. Wu, W. Wang, Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack, IEEE Trans. Comput. 62 (11) (2013) 2266–2277, http://dx.doi.org/10.1109/TC.2012.215.
[45] W. Zhang, Y. Lin, G. Qi, Catch you if you misbehave: Ranked keyword search results verification in cloud computing, IEEE Trans. Cloud Comput. 6 (1) (2018) 74–86, http://dx.doi.org/10.1109/TCC.2015.2481389.
[46] J. Shen, C. Wang, A. Wang, S. Ji, Y. Zhang, A searchable and verifiable data protection scheme for scholarly big data, IEEE Trans. Emerg. Top. Comput. 9 (1) (2021) 216–225, http://dx.doi.org/10.1109/TETC.2018.2830368.
[47] J. Cui, H. Zhou, H. Zhong, Y. Xu, AKSER: Attribute-based keyword search with efficient revocation in cloud computing, Inform. Sci. 423 (2018) 343–352, http://dx.doi.org/10.1016/j.ins.2017.09.029.
[48] J. Li, W. Yao, Y. Zhang, H. Qian, J. Han, Flexible and fine-grained attribute-based data storage in cloud computing, IEEE Trans. Serv. Comput. 10 (5) (2017) 785–796, http://dx.doi.org/10.1109/TSC.2016.2520932.
[49] J. Sun, L. Ren, S. Wang, X. Yao, Multi-keyword searchable and data verifiable attribute-based encryption scheme for cloud storage, IEEE Access 7 (2019) 66655–66667, http://dx.doi.org/10.1109/ACCESS.2019.2917772.
[50] J. Liu, Y. Li, R. Sun, Q. Pei, N. Zhang, M. Dong, V.C.M. Leung, EMK-ABSE: Efficient multi-keyword attribute-based searchable encryption scheme through cloud-edge coordination, IEEE Internet Things J. (2022) 1, http://dx.doi.org/10.1109/JIOT.2022.3163340.
[51] Q. Chen, K. Fan, K. Zhang, H. Wang, H. Li, Y. Yang, Privacy-preserving searchable encryption in the intelligent edge computing, Comput. Commun. 164 (2020) 31–41, http://dx.doi.org/10.1016/j.comcom.2020.09.012, URL https://www.sciencedirect.com/science/article/pii/S0140366420319320.
[52] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, G. Bianchi, On the design of a decentralized and multiauthority access control scheme in federated and cloud-assisted cyber-physical systems, IEEE Internet Things J. 5 (6) (2018) 5190–5204.

[53] A. Alkhulaifi, E.-S.M. El-Alfy, Exploring lattice-based post-quantum signature for JWT authentication: Review and case study, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1–5, http://dx.doi.org/10.1109/VTC2020-Spring48590.2020.9129505.

[54] D. Das, S.C. Sethuraman, S.C. Satapathy, A decentralized open web cryptographic standard, Comput. Electr. Eng. 99 (2022) 107751, http://dx.doi.org/10.1016/j.compeleceng.2022.107751, URL https://www.sciencedirect.com/science/article/pii/S004579062200060X.

[55] A.K. Ranjan, V. Kumar, M. Hussain, Security analysis of TLS authentication, in: 2014 International Conference on Contemporary Computing and Informatics (IC3I), IEEE, 2014, pp. 1356–1360.

[56] A. Ferreira, R. Giustolisi, J.-L. Huynen, V. Koenig, G. Lenzini, Studies in socio-technical security analysis: Authentication of identities with TLS certificates, in: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013, pp. 1553–1558, http://dx.doi.org/10.1109/TrustCom.2013.190.

[57] J. Zhang, L. Yang, W. Cao, Q. Wang, Formal analysis of 5G EAP-TLS authentication protocol using proverif, IEEE Access 8 (2020) 23674–23688, http://dx.doi.org/10.1109/ACCESS.2020.2969474.

[58] I. Blake, G. Seroussi, G. Seroussi, N. Smart, Elliptic Curves in Cryptography, Vol. 265, Cambridge University Press, 1999.

[59] R.C. Standard, PKCS# 1 v2. 1, RSA Laboratories, 2002, p. 61, Jun 14.

[60] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, R. Karri, CRYSTALS-Dilithium, in: Hardware Architectures for Post-Quantum Digital Signature Schemes, Springer, 2021, pp. 13–30.

[61] D.J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe, The SPHINCS+ signature framework, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2129–2146.

[62] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, S. Li, Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach, IEEE Trans. Comput. Soc. Syst. 9 (1) (2022) 134–145.

[63] H. Bangui, B. Buhnova, Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms, Comput. Electr. Eng. 100 (2022) 107901.

[64] A. Singh, K. Chatterjee, S.C. Satapathy, An edge based hybrid intrusion detection framework for mobile edge computing, Complex Intell. Syst. 8 (5) (2022) 3719–3746.

[65] A.S. Almogren, Intrusion detection in Edge-of-Things computing, J. Parallel Distrib. Comput. 137 (2020) 259–265.

[66] I. Huso, G. Piro, G. Boggia, Distributed and privacy-preserving data dissemination at the network edge via attribute-based searchable encryption, in: 2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet), 2022, pp. 122–130, http://dx.doi.org/10.1109/MedComNet55087.2022.9810394.

[67] A. Mileva, A. Velinov, L. Hartmann, S. Wendzel, W. Mazurczyk, Comprehensive analysis of MQTT 5.0 susceptibility to network covert channels, Comput. Secur. 104 (2021) 102207, http://dx.doi.org/10.1016/j.cose.2021.102207.

[68] T. Lackner, J. Hermann, F. Dietrich, C. Kuhn, M. Angos, J.L. Jooste, D. Palm, Measurement and comparison of data rate and time delay of end-devices in licensed sub-6 GHz 5G standalone non-public networks, Procedia CIRP 107 (2022) 1132–1137.

[69] M. Xu, Z. Fu, X. Ma, L. Zhang, Y. Li, F. Qian, S. Wang, K. Li, J. Yang, X. Liu, From cloud to edge: a first look at public edge platforms, in: Proceedings of the 21st ACM Internet Measurement Conference, 2021, pp. 37–53.