Distributed and Privacy-Preserving Data Dissemination at the Network Edge via Attribute-Based Searchable Encryption

Ingrid Huso, Giuseppe Piro, and Gennaro Boggia
Dept. of Electrical and Information Engineering - Politecnico di Bari, Bari, Italy
CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni
Email: {name.surname}@poliba.it

***Abstract*—Multi-Access Edge computing represents one of the most important enabling technologies for the Industrial Internet of Things. It allows advanced data processing and customized service provisioning, very close to the end-users. In the presence of many Multi-Access Edge computing applications, however, it is fundamental to ensure effective and privacy-preserving data dissemination at the network edge. From the security perspective, Attribute-based Encryption and Searchable Encryption techniques can be jointly used to achieve data confidentiality, flexible protection against unauthorized access, and privacy-preserving data dissemination. Available solutions, however, generally focus the attention on cloud-based approaches, use edge computing to implement some of the cryptographic tasks, and limit the investigation to single cryptographic operations. Indeed, no works investigate the adoption of these techniques in scenarios with multiple data producers and end-users, and fully operating at the network edge. To bridge this gap, this work proposes a novel methodology supporting fast and privacy-oriented data dissemination directly at the network edge. In the considered distributed network infrastructure, Multi-Access Edge computing applications express the interest to receive specific data by sending Trapdoors to Edge Servers. Data sources protect their contents through Attribute-based Encryption and deliver them to Edge Servers. In turn, Edge Servers implement Attribute-based Searchable Encryption functionalities to properly disseminate received contents towards Multi-Access Edge nodes hosting the applications that generated valid Trapdoors. The performance of the conceived approach has been evaluated through computer simulations. Obtained results highlight the benefits achieved against baseline (i.e., cloud-based) solutions.**

***Index Terms*—Industrial Internet of Things; Searchable Encryption; secure data dissemination; numerical analysis.**

I. Introduction

As well known, the Internet of Things (IoT) paradigm allows seamless connectivity and autonomous management in heterogeneous environments (without human interaction) and provide several important societal services via completely intelligent and automated systems [1]. The Industrial Internet of Things (IIoT), also known as Industry 4.0, further improves user experiences and promises to develop new profit streams by leveraging IoT device capabilities and data processing/analytics in the industrial domain. At the time of this writing, IIoT allows to connect smart devices and sensors to construct autonomous systems that gather, exchange, and analyze real-time data, while delivering important insights to enhance efficiency, security, and energy usage in the industry [2].
In conjunction with its development, IIoT is facing several security problems. The first one refers to the privacy protection issue. Due to the vulnerability of IIoT devices in an unsecured environment, malicious users can steal or breach sensitive data. Therefore, data must be protected through robust cryptographic techniques, directly implemented by the data provider. In this way, privacy can be guaranteed independently from the part of the network where such data will be stored (e.g., remote cloud or network edge) [3]. Secondly, the flexibility in the access control represents a critical point in data sharing because IIoT systems are no longer limited to one-to-one authorization [4]. To reduce the danger of unauthorized actions, flexible access policies are needed to regulate the accessibility and usability of services. Thirdly, differently from conventional cloud-based storage systems, upcoming IIoT deployment should deeply leverage the potentials of edge computing and the possibility to store data at the network edge (i.e., very close to the data consumers), for providing customized complex services to actuators, robots, mobile agents, controlled devices, and human workers [5].

Very promising and data-centric solutions include Attribute-Based Encryption (ABE) schemes and Searchable Encryption (SE) algorithms [6]. ABE is essential in access control because it protects data from unauthorized users [7]. SE technology is a cryptographic function able to encrypt data in a searchable manner: it allows retrieving the specific encrypted data by searching related keywords, while ensuring confidentiality [8]. Recently scientific literature presents several cryptographic schemes, where the above-mentioned techniques are combined in order to guarantee privacy-preserving solutions in file storage servers. Most of the solutions, including the ones in the IoT field proposed in [9] and [10], introduce Attribute-Based Searchable Encryption (ABSE) schemes in cloud environments where IoT devices upload encrypted documents to cloud servers and authorized users can retrieve and read them by submitting a query to the cloud, which in turn performs the search algorithm to find the required document. Anyway, from the study of the state of the art (see Section II for more details) it emerges that available solutions generally focus the attention on single cryptographic operations and propose cloud-based approaches (sometimes supported by a lightweight scheme [4] [11] or exploiting edge/fog nodes implementing part of security tasks [12] and [13]). Nevertheless, no works investigate the adoption of these techniques in scenarios with multiple data producers and end-users. Also, to the best of the authors' knowledge, the chance of sharing and disseminating data through the IIoT network in a distributed, effective, and privacy-oriented way by exploiting SE solutions represents an uncovered research goal.

To bridge this gap, this work envisages a novel methodology offering an efficient, scalable, and privacy-preserving data distribution at the network edge, by applying SE. The

reference architecture embraces heterogeneous data producers attached to a distributed network infrastructure through Network Attachment Point, Multi-Access Edge Computing (MEC) servers hosting applications, and Edge Servers (ESs). More specifically, MEC applications express the interest to receive specific data by sending Trapdoors to ESs, data producers protect their contents through ABE and send them to ESs, which implement SE to disseminate received contents only to MEC nodes hosting the applications that generated valid Trapdoors. The resulting scheme is privacy-preserving because ESs are not endowed with cryptographic material. Moreover, it registers lower dissemination delays with respect to cloud-based solutions.

The rest of the paper is organized as follows. Section II reviews the state-of-the-art on ABE and SE mechanisms. Section III illustrates the proposed methodology and provides several details about integrated cryptographic algorithms and designed communication protocol and procedures. Section IV presents an interesting numerical investigation showing the performance gains offered by the proposed approach. Finally, Section V concludes the paper and draws future research activities.

II. Related works

In the IIoT context, recent studies [6] and [14] declare that devices, networks, and application vulnerabilities are affecting everyday life and the overall industrial sector, raising the need to enhance privacy, data security, and access control. This highlights the importance of providing new methodologies able to improve the security in the data transmission flow.

A concrete solution offering fine-grained authorization, namely Attribute-Based Access Control (ABAC), has been formulated by the National Institute of Standards and Technology (NIST) [15]. The ABAC logic assumes that any resource is protected by means of dedicated access control policies, defined as a combination of properties/access grants. To access a specific resource, an end-user must prove the possession of a subset of attributes that satisfies the access control policy uniquely coupled with the resource. Some interesting cryptographic mechanisms integrate the ABAC logic directly within encryption and decryption processes. They include ABE, KP-ABE [16], and CP-ABE [17]. Indeed, these techniques can be used in the IIoT to jointly offer robust data security and flexible access control.

Regarding data dissemination, most of the available solutions (i.e., proposed in the scientific literature or implemented and ready to be used) leverage cloud-based approaches: data are distributed via remote clouds [2] [4] [11]. In these cases, however, to correctly deliver data to legitimate end-users, the server should know something about data sources, service type, end-users, and so on. If on one hand this can be an evident problem from the privacy perspective, from another hand this methodology is unfeasible in the presence of data protected with ABE.

Searchable Encryption (SE) emerges as a preliminary turning point [6]. In cloud computing environments, SE offers a useful solution for issuing search queries on encrypted files based on specific keywords. The work presented in [18] represents the first Searchable Symmetric Encryption (SSE) scheme where the symmetric key encryption method is used to build the searchable ciphertexts and to allow users to generate trapdoors through the shared key. Later, the contribution in [19] integrates the keyword searching with public key encryption techniques, allowing users to securely recover the requested files over encrypted data using user-defined keywords. The Public-Key Searchable Encryption (PKSE) works with both public and private key enabling data owners and users to do encryption with their public keys and produce trapdoors with their private keys. Following that, the scientific literature presents numerous PKSE systems with various capabilities, such as single keyword search [20] [21], fuzzy keyword search [22], verified keyword search [23], and ranked keyword search [24].

However, the mentioned SE systems do not allow data owners to give end-users fine-grained search capabilities. Indeed, studies [25] and [26] have recently looked at the integration of ABE and SE systems. Nevertheless, these approaches can only be utilized to find a particular keyword, limiting the flexibility and accuracy of data retrieval. Thus, works in [27], [28], and [29] have also looked into attribute-based multi-keyword search algorithms. Moreover, [30] suggests an enhanced ABE method with multi-keyword search to facilitate simultaneous numeric attribute comparison, hence significantly increasing the flexibility of ABE encryption in a dynamic IoT context.

As far as IoT is concerned, novel lightweight SE approaches are proposed in edge and fog computing environments since there are considered promising solutions able to bring data storage and computation capabilities closer to IoT devices [31]. Indeed, [32] envisages a dynamical SE process with multi-keyword search for smart grids in a cloud-edge architecture where the search algorithm is running through a cooperation between the edge nodes and the cloud server. While, the studies in [33], [7], and [34] introduce three different SE schemes in fog-based IoT scenarios, where fog nodes both help the cloud in the searching and forwarding process and partially decrypt the retrieved documents in order to reduce the computational workload on IoT devices.
Recently, the work in [13] introduces a distributed SE scheme in the healthcare domain, demonstrating the capability of fog nodes to decrease the computational workflow with respect to the cloud environment. Nevertheless, it provides fog nodes with cryptographic capabilities to partially decrypt and encrypt searched queries.

The analysis of related works demonstrates that all the presented studies focus the attention only on the cryptographic feature of the SE schemes, with the target goal of identifying (i.e., search and retrieve) specific encrypted files. Indeed, it is possible to summarize some open issues, interesting for the scientific community, as in what follows:

- 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

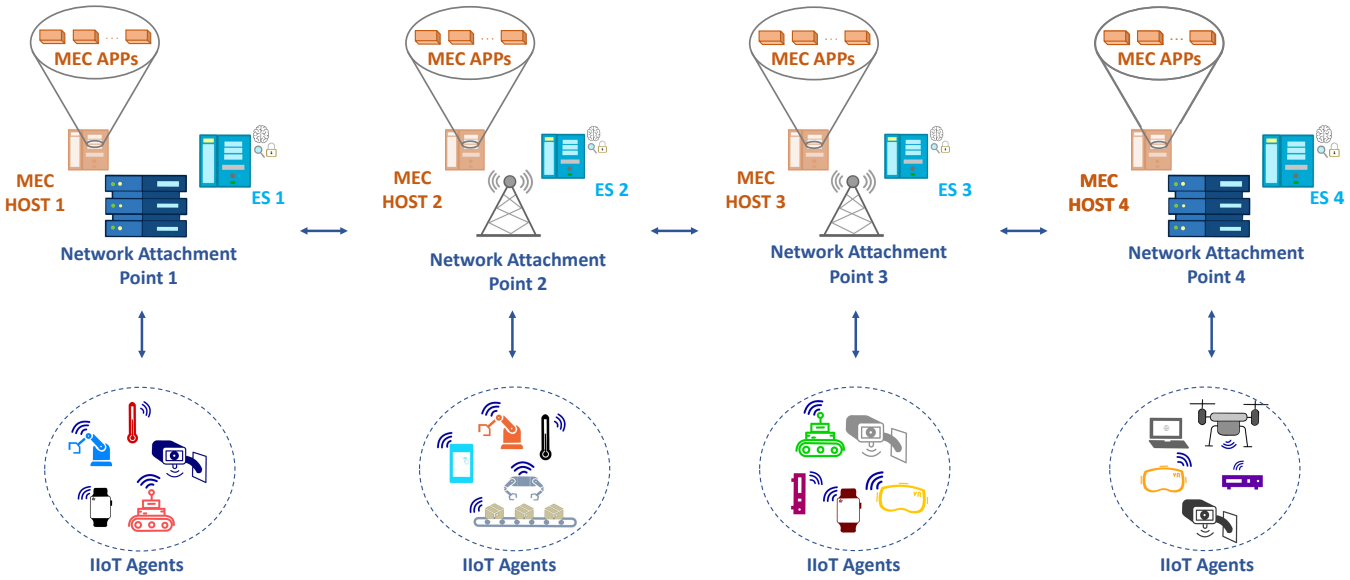


Fig. 1. The reference distributed network architecture.

constrained devices. However, the chance of performing SE operations directly at the edge of the network has not yet been investigated.

- 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.

III. The Conceived Data Dissemination Scheme

Fig. 1 depicts the distributed network architecture considered in this work. Here, heterogeneous Network Attachment Points offer wireless or wired connectivity to groups of IIoT agents (i.e., sensors, wearables, fixed robotic arms, mobile robots, drones, and industrial processes). MEC hosts and ESs are deployed at the edge of the network. According to ETSI-MEC specifications [35], each MEC host may integrates multiple MEC applications that, by exploiting powerful tools and computational resources, are able to process (e.g, data mining and fusion) heterogeneous data generated by IIoT agents, as well as to provide advanced and specialized services close to the end-users. On the other hand, ESs are in charge of processing the received traffic flow, while routing and forwarding data at the network edge. Without loss of generality, it is assumed that beyond each Network Attachment Point are available one ES and one MEC host (handling many MEC applications). As anticipated in the introduction, data dissemination is autonomously handled by the ES, via ABSE.

A. Design Principles

The conceived approach leverage the following design principles.

First of all, security is enforced by an Authority that is a fully trusted third party responsible for the system setup. Specifically, it deals with system security initialization parameters, key material generation, attribute management, and policy enforcement.

MEC applications, which are in possession of a precise set of attributes (generated and released by the aforementioned trusted Authority), request data identified with a set of *keywords*. For example, a monitoring application can be interested to know the variables measured by all the available sensors, an AR/VR application is interested to retrieve all the data associated with a given industrial process, an indoor navigation process needs to know the location of robots and packages, and so on. According to the ABSE scheme (whose technical details are presented in the next sub-section), each request is encoded via *search Trapdoors*, based on the selected keywords and attributes. Each MEC host collects the Trapdoors generated by its MEC applications and shares them with all the available ESs. Since Trapdoors hide the search keywords and attributes through cryptographic operations, ESs can not retrieve any information about application interests and related access capabilities (i.e., privacy-oriented approach).

IIoT agents generate data (e.g., multimedia contents, time-series values, and so on) and outsource them to the closest ES. In other words, they represent the data producers. To this end, they select the specific keywords associated with the generated data, encrypt both keywords and data through ABE, and deliver the overall output to the closest ES.

Each ES handles a *Trapdoor table*, which jointly stores application requests and reference MEC host. Note that the Trapdoor table is a completely new entity envisaged in this contribution, and properly used to distribute data directly at the network edge, in an effective, distributed, and privacy-oriented way. ESs have a twofold contribution: i) running the *search algorithm* over encrypted data, and ii) *disseminating data* towards specific MEC hosts. Therefore, when a new data is received, ES scrolls the Trapdoor table in order to find the Trapdoors that match the keywords and the policies defined in the encrypted data. The search procedure returns the list of MEC hosts which previously sent valid MEC applications trapdoors that matches both keywords and policies of data producers. Also in this case, it is worth mentioning that the search procedure does not provide any meaningful information on the search content to ESs. Accordingly, the resulting approach ensures a privacy-preserving behavior: elements at the network edge receive and distribute data without revealing the related contents, since SE is used.

B. Technical details about the data dissemination workflow

This subsection formalizes both search and data dissemination processes, by providing technical details about security operations to be implemented. It is very important to remark that this contribution does not propose a novel ABSE algorithm, but it aims at integrating one of the techniques already provided in the current scientific literature for supporting a fast and privacy-oriented data dissemination at the network edge. As a consequence, any ABSE mechanism can be integrated within the overall data dissemination workflow discussed herein. However, without loss of generality, the ABSE algorithm presented in [9] is taken as a reference example, since it has been found to be less computationally expensive than others, and the overall complexity remains constant even as the number of users' attributes increases.

The overall data dissemination workflow is divided into five distinct phases, illustrated in Fig. 2 and detailed below.

Phase 1: system setup. The selected ABSE 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:

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

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 MEC applications 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 MEC application, the Authority produces the secret key for that application. Basically, a MEC application that joins the industrial network sends its set of attributes $\mathrm{X} = (x_1, x_2, ..., x_n)$ 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 MEC application, S_k, is computed as:

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

and shared with the reference application.

Phase 2: trapdoor generation and forwarding. During this phase, the MEC application generates the search Trapdoor, that is t_Φ. Specifically, starting from its secret key S_k, the set of k keywords $\Phi = (\phi_1, \phi_2, ..., \phi_k)$ of its interest, and a random number $z_p \in \mathbb{Z}_p^*$, the Trapdoor is calculated as:

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

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 $td_3 = h_2^{z_p}$.

As anticipated in the previous sub-section, the Trapdoor is shared with all the ESs in the system.

Phase 3: encryption and outsourcing. Let M be the data to encrypt and outsource to the ES. $\Psi = (\psi_1, \psi_2, ..., \psi_z)$ 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:

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

where:

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

Finally, the data producer sends the ciphertext ct to the reference ES.

Phase 4: search and data forwarding. This phase involves the ES, which performs the search algorithm to determine whether the received encrypted data matches one or more queries stored into the Trapdoor table. *Differently from the current scientific literature*, the procedure proposed herein operates in a scenario with multiple IIoT agents and multiple MEC applications. In details, for each received data ct and for each stored Trapdoor t_Φ, the ES verifies that the following equation holds:

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

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 S belonging to the MEC application 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.

During the search algorithm, all the Trapdoors are processed. However, if multiple Trapdoors received from the same MEC host produce a match, the ES delivers the encrypted data to that MEC host only once, denoting the list of interested MEC applications. In this way, the proposed approach also ensures a reduction in bandwidth consumption.

For sake of clarity, search and data forward operations are defined in the Algorithm 1.

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

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

C. Running Example

This section, presents a running example willing to better explain operations and the interactions to be performed. The use case scenario considers 3 Network Attachment Points equipped with a MEC host and an ES. Moreover, as illustrated in Fig. 3, each ES has a Trapdoor table where all receiving Trapdoors are stored and listed with respect to the referred MEC host. The example use case considers an AR/VR application and a sensing control as MEC applications and a

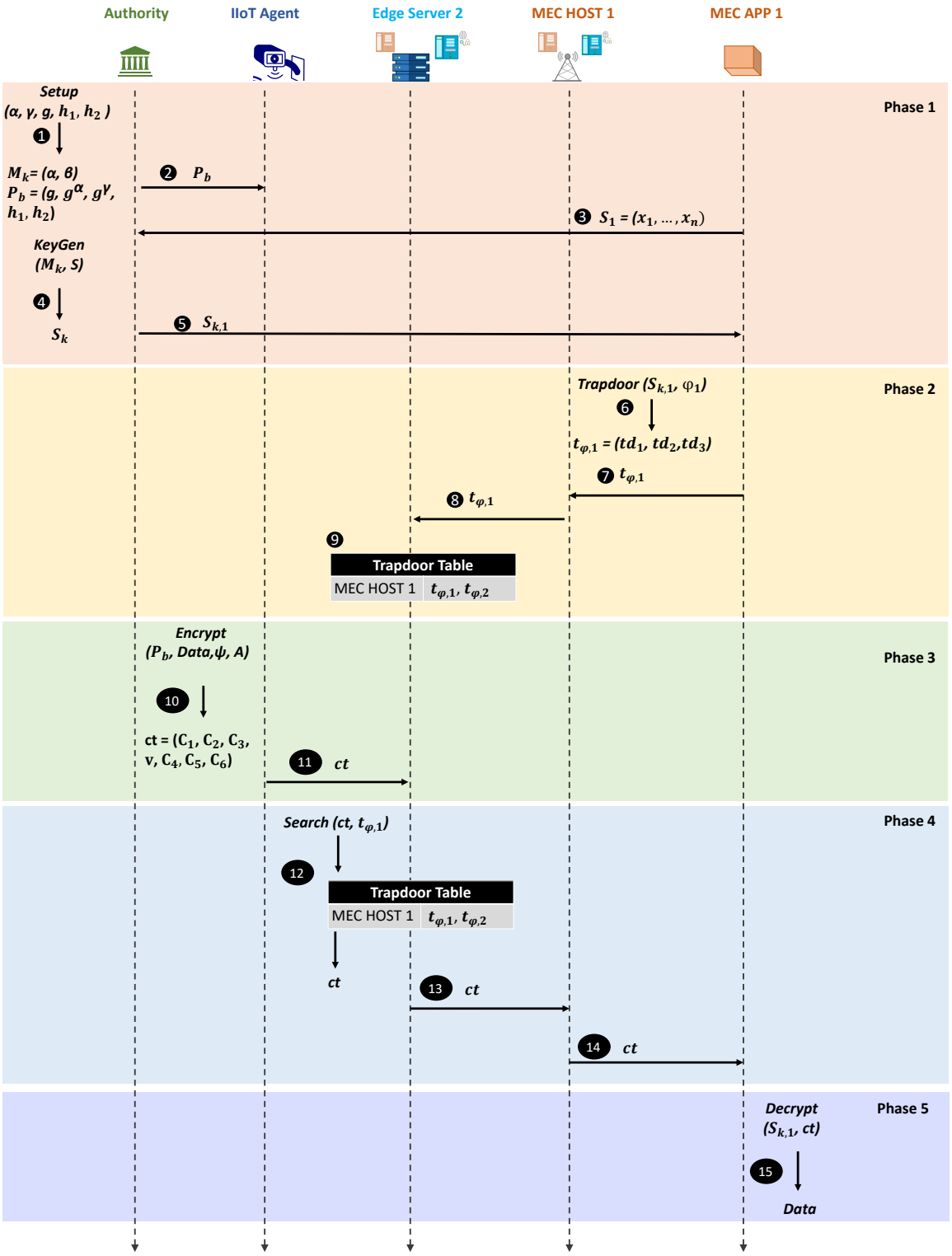


Fig. 2. Data dissemination workflow.

mobile robot with an integrated camera as a data producer. Specifically, the MEC host 1, holds two MEC applications: an AR/VR application and a sensing control one. These two generate query Trapdoors and the MEC host 1 forwards them to all the ESs. Lets assume that both Trapdoors contain ”AR”, ”video”, and ”robot” as query keywords and that they are asking for a video stream flow outsourced from a mobile robot with an integrated camera, who is attached at the ES 2. When the mobile robot outsources the data to the ES 2, this one checks within its Trapdoor table by comparing the encrypted data with the encrypted query keywords. As soon as it finds the matching Trapdoor (i.e, the one of the AR/VR application) it sends the data to the MEC host 1 and records that the data has been sent to that MEC host. In this way, when the Trapdoor referred to the sensing control application matches the ciphertext, the ES 2 does not re-send the same data to the same MEC host.

IV. Numerical Evaluation

To demonstrate the great potentials of the conceived privacy-preserving data dissemination scheme, this section

Algorithm 1 The proposed search and data forwarding phase

Each MEC application sends $t_{\phi,i} = (td_1, td_2, td_3)$ to the MEC host
The MEC host forwards $t_{\phi,i}$ to ESs
Each ES stores $t_{\phi,i}$ in its Trapdoor table
The ES receives $ct = (C_1, C_2, C_3, v, C_4, C_5, C_6)$ from a data producer
For each MEC host registered into the ES Trapdoor table
while *Search*$(ct,\ t_{\phi,i})$ = 0 **do**
  if $e(C_4, td_1) \cdot C_5^{td_2} = e(C_6, td_3)$ **then**
    Search(ct,$t_{\phi,i}$) = 1
    if ct has not been sent to the MEC host **then**
      the ES forwards ct to the MEC host
      The ES records that ct is sent to the MEC host
    end if
  else
    Search(ct,$t_{\phi,i}$) = 0
  end if
end while

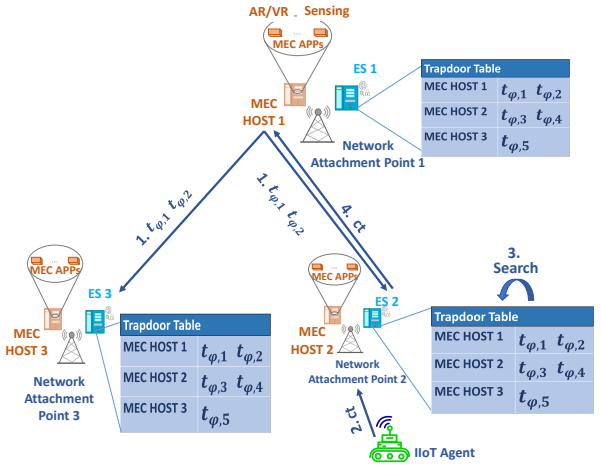


Fig. 3. Running example.

presents a numerical investigation conducted in different scenarios. To this end, a MATLAB script has been developed to model a distributed IIoT environment. The investigated KPIs include: i) the *average search time*, defined as the amount of time the node implementing the Searchable Encryption algorithm takes to check the received data with all the available Trapdoors, and ii) the *average delivery delay*, expressed as the average amount of time needed to deliver the generated data to the MEC applications that issued the right Trapdoors. Results are compared against those registered by a baseline approach, where data and Trapdoors are managed by a remote cloud (which performs, in a centralized way, searching and delivery tasks).

The study considers a network with a variable number of Network Attachment Points, ranging from 2 to 10. Indeed, the number of MEC hosts and ESs available in the considered network infrastructure ranges from 2 to 10, as well. Let N_{es} be the number of available ES.

A variable number of data producers, N_{dp}, are randomly and uniformly distributed among network cells served by the aforementioned Network Attachment Points. Specifically, N_{dp} is set to 10, 50, or 100. Without loss of generality, it is assumed that these devices generate data for S different

TABLE I
COMPUTATIONAL COST OF CRYPTOGRAPHIC OPERATIONS.

Cryptographic operation	Execution time [ms]
Pairing in $\mathbb{G}$ (P)	27.98
Exponentiation in $\mathbb{G}$ (E)	18.62
Exponentiation in $\mathbb{Z}_p$ (E_z)	0.759
Multiplicative in $\mathbb{Z}_p$ (M_z)	0.0058385
Search Time	$T_{se} = 1 * E + 2 * P =$ $= 74.58$
Encryption	$T_{enc} = 3 * P + 8 * E + 2nM_z =$ $= 232.9 + 2 * n * 0.0058385$
Decryption	$T_{dec} = 1 * P + 1 * E + 2 * M_z =$ $= 46.61$

services. Therefore, these data are protected according to the access policies configured for the type of service they belong to. Each test randomly maps a data producer to one of the available service types. Moreover, the access policy is defined through a combination of n attributes.

On the other hand, a total number of MEC applications, N_{app}, are randomly and uniformly distributed among the available MEC hosts. In line with the previous assumptions, each test randomly maps a MEC application to one of the available service types. Thus, each MEC application is configured to request (via Trapdoors and according to the protocol discussed in the previous Section) all the data belonging to a given service type. The number of MEC applications is chosen in the range from 20 to 100.

A. Analysis of the average search time

The study has been conducted by using a windows system with Intel Core i7 CPU at 2.60GHz. According to [9], the computational cost associated with a single search operation in the ABSE scheme considers one pairing in $\mathbb{G}$ (P), one Exponentiation in $\mathbb{G}$ (E), and one Multiplicative in $\mathbb{Z}_p$ (M_z). The resulting search time, namely T_{se} is reported in table I.

The scientific literature also demonstrated that the amount of time to perform multiple search operations linearly increases with the number of generated data or Trapdoors to be checked (see [36] for example). Indeed, given the number of MEC applications N_{app}, the number of data producers managed by the $i-th$ ES N_{dp}^i, and by assuming that all these data producers generate data within the same observation time interval (worst case), the resulting search time is equal to:

$$\hat{T_{se}}\Big|_{proposal} = T_{se} + \beta(N_{dp}^i N_{app} - 1). \quad (6)$$

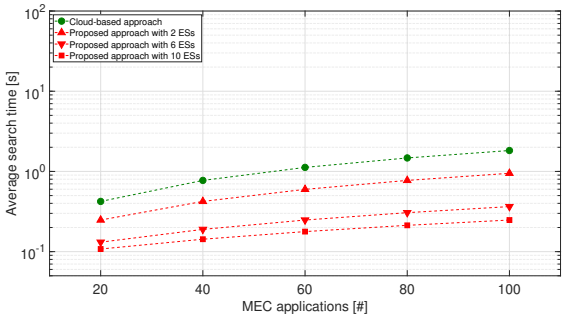
Indeed, when a single data producer (e.g., $N_{dp}^i = 1$) and one MEC application (e.g., $N_{app} = 1$) are considered, the search time results in:

$$\hat{T_{se}}\Big|_{proposal} = T_{se}.$$

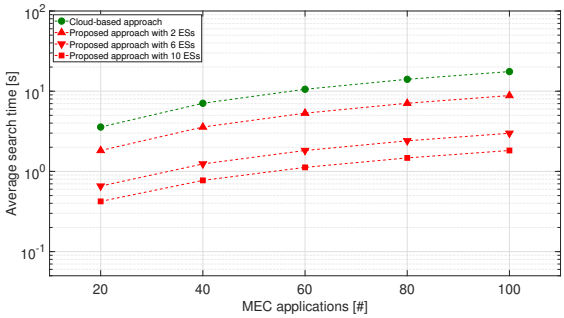
Now, considering that the average number of data producers managed by the $i-th$ ES is equal to $\bar{N}_{dp} = E[N_{dp}^i] = N_{dp}/N_{ES}$, the average search time achievable by each single ES is equal to:

$$\bar{T}_{se}\Big|_{proposal} = E[\hat{T_{se}}] = T_{se} + \beta\left(\frac{N_{dp}N_{app}}{N_{SE}} - 1\right). \quad (7)$$

A different story is experienced for the cloud-based approach. In this case, in fact, the search operation is performed



(a) Scenario with 10 active data producers.



(b) Scenario with 100 active data producers.

Fig. 4. Average search time vs number of MEC applications.

only in a single node of the network. Therefore, given the number of MEC applications N_{app}, the total number of data producers N_{dp}, and by assuming that all these data producers generate data within the same observation time interval (worst case), the resulting average search time is equal to:

$$\bar{T}_{se}\Big|_{cloud} = T_{se} + \beta(N_{dp}N_{app} - 1). \tag{8}$$

The analysis of the state of the art suggests to setting $\beta = 1.75$ [36]. Based on these premises, the average search time achievable in different scenarios is reported in Fig. 4. Reported results demonstrate that the average search time of both approaches increases with the number of data producers and MEC applications. The proposed approach, by distributing the search procedure on different ESs on the edge of the network, permits to obtain a shorter search process time. Indeed, by considering the scenario with 100 data producers and 10 MEC applications, the average search time is reduced by about 81.6% with 6 ESs and about 88.2% with 10 ESs. In the same way, by introducing 100 data producers and 100 MEC applications, the search time decreases by about 83% with 6 ESs and about 89.6% with 10 ESs. Finally, focusing the attention on 100 data producers, it could be noticed how passing from 10 to 100 MEC applications the average search time increases of 14 seconds for the cloud and 1.4 seconds for the proposed approach with 10 ESs.

B. Analysis of the average delivery delay

In order to estimate the average delivery delay, it is necessary to introduce the following variables: i) $\bar{T}_{radio}$, that represents the delivery delay in the radio interface, ii) $\bar{T}_{edge}$, that represents the delivery delay at the network edge, and iii) $\bar{T}_{cloud}$, that represents the delivery delay experienced when a remote cloud is contacted. These variables have been evaluated through Ping and Trace Route tests. In particular,

TABLE II
AVERAGE COMMUNICATION DELAYS

Communication type	Average RTT [ms]	Delay [ms]	Number of hops [#]
$\bar{T}_{radio}$	1.309	0.6545	1
$\bar{T}_{edge}$	16.594	8.297	1
$\bar{T}_{cloud}$	42.08	21.04	22

by using a computer connected to the network of Politecnico di Bari, for each test the average RTT on 10^3 consecutive pings has been considered. First, a test on a remote amazon server has been made to estimate the communication delay between a data producer and the remote cloud server (i.e., $\bar{T}_{cloud}$). The second test has been done for evaluating the average communication delay experienced for contacting the closest Network Attachment Point (i.e., $\bar{T}_{radio}$). Finally, a test on another device connected to the same network at the Politecnico di Bari has been done to estimate the communication time at the edge (i.e., $\bar{T}_{edge}$). Results, used to evaluate the average delivery delay, are reported in Table II.

The analysis of the average delivery delay should also consider the amount of time required to encrypt and decrypt data, denoted with T_{enc} and T_{dec}, respectively. Their values are reported in Table I.

Regarding the methodology proposed in this work, the average delivery delay can be evaluated as:

$$\begin{aligned}\bar{T}_{del}\Big|_{proposal} &= T_{enc} + \bar{T}_{radio} + \\ &+ \left[T_{se} + \beta\left(\frac{N_{dp}N_{app}}{N_{SE}} - 1\right)\right] + \\ &+ \bar{T}_{edge} + T_{dec}.\end{aligned} \tag{9}$$

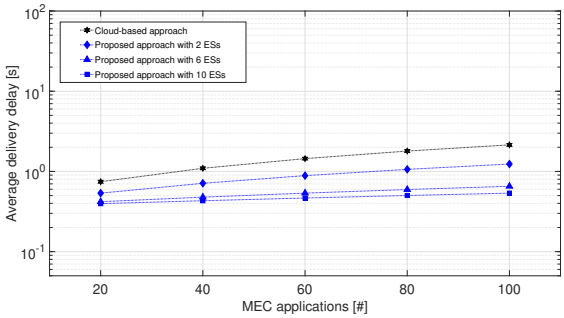
Differently, in the cloud-based solution, the average delivery delay can be evaluated as:

$$\begin{aligned}\bar{T}_{del}\Big|_{cloud} &= T_{enc} + 2\bar{T}_{cloud} + \\ &+ \left[T_{se} + \beta\Big(N_{dp}N_{app} - 1\Big)\right] + \\ &+ T_{dec},\end{aligned} \tag{10}$$

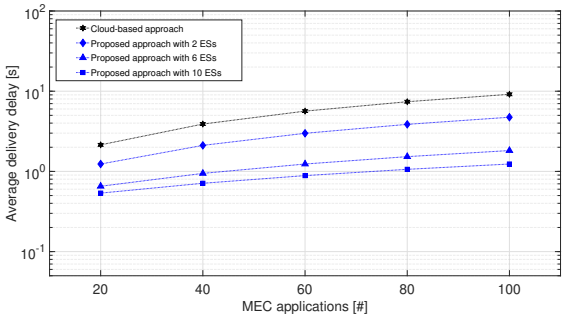
where, $2\bar{T}_{cloud}$ is the sum of the time needed to send the data to the cloud server and the time required to the deliver the searched ciphertext back to the MEC host.

Fig. 5 shows the average delivery delay as a function of the number of MEC applications and ESs. The three sub-figures refer to scenarios with different number of data producers. Results highlight how distributing search operations on the edge of the network allow decreasing the amount of time needed for retrieving the query data flow. Indeed, as the number of data producers increases, the distance between the average delivery delay of the cloud-based approach and the proposed one increases, passing from a difference of a few seconds with 10 data producers to a difference of about 10 seconds with 100 data producers. Thus, the deployment of search operation directly on the edge of the network allows reducing the average delivery delay up to 45% with respect to the baseline approach.

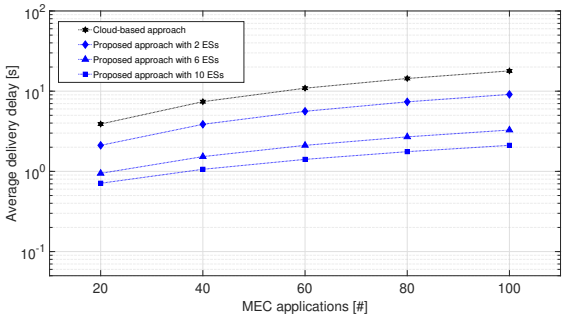
Finally, since table I shows that the number of attributes affects the encryption time, Fig. 6 reports the evaluation of



(a) Scenario with 10 active data producers.

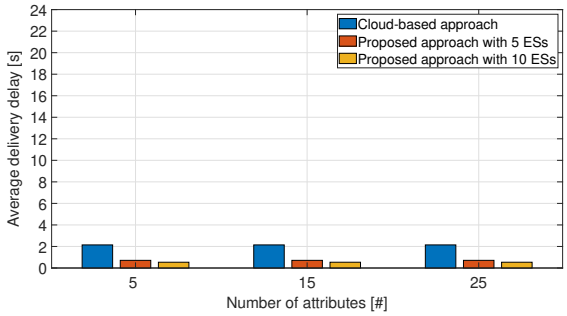


(b) Scenario with 50 active data producers.

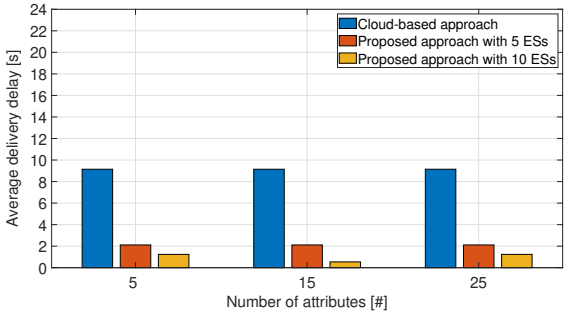


(c) Scenario with 100 active data producers.

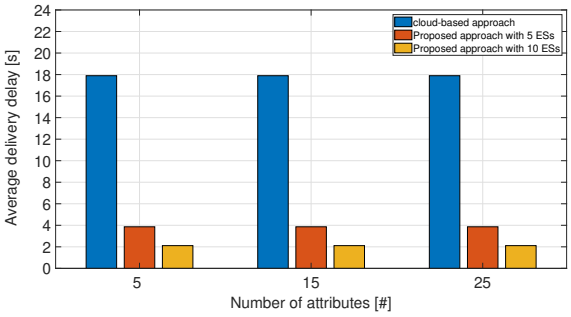
Fig. 5. Average delivery delay vs number of MEC applications.



(a) Average delivery delay with 10 MEC applications



(b) Average delivery delay with 50 MEC applications



(c) Average delivery delay with 100 MEC applications

Fig. 6. Average delivery delay vs numbers of attributes.

the average delivery delay with different attributes. In line with [9], it can be noticed that, by fixing the number of data producers to 100, the variation of the number of attributes causes a marginal change in the average delivery delay, while increasing the number of MEC applications.

V. Conclusions

This paper proposed a distributed and privacy-preserving data dissemination mechanism, which operates at the network edge and leverages Attribute-based Searchable Encryption. The reference architecture embraces heterogeneous data producers attached to a distributed network infrastructure through Network Attachment Point, Multi-Access Edge computing servers hosting applications, and Edge Servers. More specifically, Multi-Access Edge computing applications express the interest to receive specific data by sending Trapdoors to Edge Servers, data producers protect their contents through Attribute-based Encryption and send them to Edge Servers, and Edge Servers implement Searchable Encryption functionalities and disseminate received contents only to Multi-Access Edge nodes hosting the applications that generated valid Trapdoors. The numerical analysis demonstrated that the proposed approach ensures a lower computational complexity with respect to cloud-based solutions, thus offering lower dissemination delays. Future research activities intend to further investigate the performance of the proposed approach in a more realistic scenario (hence, considering different statistics in the data generation process), while also evaluating bandwidth and energy consumption. At the same time, they also intend to formulate an optimized algorithm willing to distribute Edge Servers at the network edge, based on traffic load, heterogeneous computational and communication requirements, and users dynamicity.

Acknowledgements

This work was supported by the PRIN project no. 2017NS9FEY entitled “Realtime Control of 5G Wireless Networks: Taming the Complexity of Future Transmis- sion and Computation Challenges” funded by the Italian MIUR, as well as in the context of the GUARD project, which receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement 833456. It has been also supported by the Italian MIUR PON projects Pico&Pro (ARS01 01061), AGREED (ARS01 00254), FURTHER (ARS01 01283), RAFAEL (ARS01 00305).

REFERENCES

[1] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022.

[2] S. Li, S. Zhao, G. Min, L. Qi, and G. Liu, "Lightweight privacy-preserving scheme using homomorphic encryption in industrial internet of things," *IEEE Internet of Things Journal*, 2021.

[3] A. Ometov, O. L. Molua, M. Komarov, and J. Nurmi, "A survey of security in cloud, edge, and fog computing," *Sensors*, vol. 22, no. 3, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/927

[4] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, and M. Imran, "Lightweight searchable encryption protocol for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4248–4259, 2021.

[5] R. W. Coutinho and A. Boukerche, "Design of edge computing for 5g-enabled tactile internet-based industrial applications," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 60–66, 2022.

[6] T. Soo Fun and A. Samsudin, "Recent technologies, security countermeasure and ongoing challenges of industrial internet of things (iiot): A survey," *Sensors*, vol. 21, no. 19, 2021.

[7] H. Li and T. Jing, "A lightweight fine-grained searchable encryption scheme in fog-based healthcare iot networks," *Wireless Communications and Mobile Computing*, 2019.

[8] N. Andola, R. Gahlot, V. K. Yadav, S. Venkatesan, and S. Verma, "Searchable encryption on the cloud: a survey," *The Journal of Supercomputing*, pp. 1–33, 2022.

[9] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh, and X. Liu, "Secure fine-grained encrypted keyword search for e-healthcare cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1307–1319, 2021.

[10] T. Liu, Y. Miao, K.-K. R. Choo, H. Li, X. Liu, X. Meng, and R. H. Deng, "Time-controlled hierarchical multi-keyword search over encrypted data in cloud-assisted iot," *IEEE Internet of Things Journal*, 2021.

[11] B. Chen, L. Wu, N. Kumar, K.-K. R. Choo, and D. He, "Lightweight searchable public-key encryption with forward privacy over iiot outsourced data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 4, pp. 1753–1764, 2021.

[12] Y. Tao, P. Xu, and H. Jin, "Secure data sharing and search for cloud-edge-collaborative storage," *IEEE Access*, vol. 8, pp. 15 963–15 972, 2020.

[13] Mamta, B. B. Gupta, and M. D. Lytras, "Fog-enabled secure and efficient fine-grained searchable data sharing and management scheme for iot-based healthcare systems," *IEEE Transactions on Engineering Management*, pp. 1–13, 2022.

[14] V. Pedreira, D. Barros, and P. Pinto, "A review of attacks, vulnerabilities, and defenses in industry 4.0 with new challenges on data sovereignty ahead," *Sensors*, vol. 21, no. 15, 2021.

[15] *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, NIST, NIST Special Publication 800-162, 2014.

[16] C.-J. Wang and J.-F. Luo, "A key-policy attribute-based encryption scheme with constant size ciphertext," in *2012 Eighth International Conference on Computational Intelligence and Security*, 2012.

[17] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007.

[18] D. X. Song, D. Wagner, and 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.

[19] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds. Springer Berlin Heidelberg, 2004, pp. 506–522.

[20] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing peks schemes secure against keyword guessing attacks is possible?" *Computer communications*, vol. 32, no. 2, pp. 394–396, 2009.

[21] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, 2016.

[22] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

[23] J. Shen, C. Wang, A. Wang, S. Ji, and Y. Zhang, "A searchable and verifiable data protection scheme for scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 216–225, 2021.

[24] W. Zhang, Y. Lin, and G. Qi, "Catch you if you misbehave: Ranked keyword search results verification in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 74–86, 2018.

[25] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1981–1992, 2015.

[26] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "Akser: Attribute-based keyword search with efficient revocation in cloud computing," *Information Sciences*, vol. 423, pp. 343–352, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025516311215

[27] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.

[28] Y. Miao, X. Liu, R. H. Deng, H. Wu, H. Li, J. Li, and D. Wu, "Hybrid keyword-field search with efficient key management for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3206–3217, 2019.

[29] J. Sun, L. Ren, S. Wang, and X. Yao, "Multi-keyword searchable and data verifiable attribute-based encryption scheme for cloud storage," *IEEE Access*, vol. 7, pp. 66 655–66 667, 2019.

[30] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018.

[31] N. Karnik, U. Bora, K. Bhadri, P. Kadambi, and P. Dhatrak, "A comprehensive study on current and future trends towards the characteristics and enablers of industry 4.0," *Journal of Industrial Information Integration*, p. 100294, 2021.

[32] K. Fan, Q. Chen, R. Su, K. Zhang, H. Wang, H. Li, and Y. Yang, "Msiap: A dynamic searchable encryption for privacy-protection on smart grid with cloud-edge-end," *IEEE Transactions on Cloud Computing*, 2021.

[33] R. Zhou, X. Zhang, X. Wang, G. Yang, H. Wang, and Y. Wu, "Privacy-preserving data search with fine-grained dynamic search right management in fog-assisted internet of things," *Information Sciences*, vol. 491, pp. 251–264, 2019.

[34] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 772–785, 2019.

[35] *Multi-access Edge Computing (MEC): Framework and Reference Architecture*, ETSI, ETSI GS MEC 003 v.2.1.1, 2019.

[36] M. Wang, Y. Miao, Y. Guo, C. Wang, H. Huang, and X. Jia, "Attribute-based encrypted search for multi-owner and multi-user model," in *ICC 2021 - IEEE International Conference on Communications*, 2021.