# Lightweight Privacy-Preserving Scheme Using Homomorphic Encryption in Industrial Internet of Things

Shancang Li, Shanshan Zhao, Geyong Min, *Member, IEEE*, Lianyong Qi, and Gang Liu

***Abstract*—The emerging technologies, such as *smart sensors, 5G/6G wireless communication, artificial intelligence, etc.*, have been maturing the future Internet of Things (IoT) by connecting the massive number of devices, which are expected to consistently collect and transmit real-time data to support business intelligence in an efficient and privacy-preserving way. The IoT can afford businesses predictive maintenance, improve field service, asset tracking, and further enhance customer satisfaction and facility management in industrial sectors. However, the privacy concern in IoT is a big challenge in IoT applications and services. This work proposed a lightweight privacy-preserving scheme based on homomorphic encryption in the context of the IoT, in which we investigated and analyzed the privacy issues between the data owners, untrustworthy third-party cloud servers, and the data users. Meanwhile, computationally efficient homomorphic algorithms are proposed to guarantee the privacy protection for the data users. Experimental results demonstrate that the proposed scheme can effectively prevent privacy breaches in IoT.**

***Index Terms*—Industrial Internet of Things (IIoT), IoT security, lightweight privacy, provenance, security.**

## I. Introduction

THE Internet of Things (IoT) technology is being broadly used and can have a huge impact in our daily lives [1]. In an industry setting, the Industrial IoT (IIoT), or Industry 4.0, is expected to improve the user experiences and create new business streams, taking advantage of the new capability of the IIoT device and secure data analytics [2], [3]. The IIoT connects smart machines and sensors to form automatic systems that collect, exchange, and analyze real-time data, and deliver valuable insights to improve the performance, safety, reliability, and energy consumption of industry sectors [1], [4]–[6].

The IIoT is promising in many industrial areas, including healthcare, smart manufacturing, smart city, smart grid, etc. As a vast and more complex system, the IIoT is expected to offer huge potential benefits to existing industry systems by automating projects, optimizing digital transformation goals, improving productivity, reducing costs, etc. Meanwhile, the IIoT is expected to make a far-reaching impact on the operation of industries around the world. The IIoT can provide business predictive maintenance, which uses real-time data to accurately predict defects in smart devices, and enabling industrial sectors to take action to address those issues before apart fails or a device goes down. The IIoT can improve field service by identifying potential issues in industry systems before they become major problems. It can also provide asset tracking to monitor the real-time location, conditions, and storages, which is important in healthcare, supply chain, manufacturers, logistics, etc. [1], [7].

The IIoT connects industry devices to the Internet and can bring many benefits, however, it also leaves devices in IIoT vulnerable to hacking and solutions to prevent IIoT devices from being exploited are necessary. In many IIoT applications, such as healthcare, life-threatening cyberattacks are targeting medical devices (such as insulin pumps attack and baby monitor) and can cause serious security issues [8], [9]. Actually, the IIoT systems are facing many challenges and becoming the targets of cyberattacks, as summarized below.

*Security in IIoT:* In the past few years, a number of security solutions have been proposed for IIoT, including cryptographic techniques, data encryption algorithms, secure communication channels, strong anonymous authentication, and access controls in IIoT. A number of promising technologies, such as attributed-based access control, or even blockchain technologies have been developed in ensuring IIoT security.

*Privacy in IIoT:* The information leakage is a big concern in IIoT in many critical IIoT systems, such as smart grid, industrial critical systems (ICS)/supervisory control and data acquisition (SCADA) systems, etc., in which the need for privacy assurance of data collected, as well as the privacy issues associated with critical infrastructures and IoT services, has been raised in the literature. The data aggregation in key IIoT applications is the core of privacy preserving [10], [11].

*Safety:* Many IIoT devices suffer from intermittent defects that can cause device on fault. In the safety of IIoT, the following issues need to be clearly addressed: device malfunction, device communications, device incompatibilities and errors, unintended accident, malicious intents, etc.

This article aims at developing a secure solution by balancing the security and efficiency in IIoT. A lightweight

Manuscript received 30 August 2020; revised 14 January 2021; accepted 28 February 2021. Date of publication 24 March 2021; date of current version 8 August 2022. *(Corresponding author: Shancang Li.)*

Shancang Li is with the School of Computer Science and Informatics, Cardiff University, Cardiff CF24 3AA, U.K. (e-mail: Shancang.Li@ieee.org).

Shanshan Zhao is with the Department of Engineering Design and Mathematics, University of the West of England, Bristol BS16 1QY, U.K.

Geyong Min is with the Department of Computer Science, University of Exeter, Exeter EX4 4PY, U.K.

Lianyong Qi is with the School of Information Science and Engineering, Qufu Normal University, Qufu 56650, China.

Gang Liu is with the School of Computer Science and Technology, Xidian University, Xi'an 710072, China.

Digital Object Identifier 10.1109/JIOT.2021.3066427

privacy-preserving scheme will be detailed for the IIoT system. An air quality monitoring use case will be given to address the challenges mentioned above. The main contributions of this article include the following.

1) This work proposes a secure data processing framework for applications in IIoT that well address the privacy issues between data owners, untrusted cloud server, and the data users, and a data labeling scheme in IoT environment for resource-constrained devices, which allows data users to learn a label $\ell_i$ at each node $i$ while keeping low communication complexity. A label is typically a unique index of the data created by a node and could be designed as a small size integer.
2) A lightweight privacy-preserving scheme for both data owner and data user in IIoT is proposed that by dividing computational costs into a fixed and a dynamic part, and in data processing only dynamic part is updated.
3) We improved upon the labHE of [12] by leveraging a preprocessing phase, where the computational cost is reduced. The data users (applications) no longer need to perform an expensive evaluation on the resulting ciphertexts. This allows apps on IoT devices to utilize more efficient homomorphic encryption (HE) and further improve the performance.

The remainder of this work is organized as follows. Section II elaborates the recent works in the field of lightweight privacy-preserving in IIoT; and Sections III and IV present the proposed lightweight privacy-preserving scheme for IIoT systems in detail. A use case is introduced in Section V. Finally, Section VI concludes this work.

## II. Related Works

The privacy preserving at endpoint devices in IIoT has been discussed as a core security issue in the past few years [1]. The privacy issues in the IoT have been well studied and a number of privacy-preserving protocols have been developed for IoT devices and applications [1], [4]. With the advent of new techniques in the industry, the privacy concerns in IIoT (or Industry 4.0) are on the rise as well [1]. The IIoT is able to generate, collect, store, and process huge amounts of data, it has become a main target of cyberattacks that can cause large-scale system failures and massive destruction [6]. The security and privacy challenges in IIoT systems are very complex than the existing industry automation systems [13], in which data generated by IIoT device must be protected against cyber attacks. Efficient ways are needed to identify, secure remote remediation [2].

In general, an IIoT system consists of a large number of IIoT devices, which vary widely in terms of computational capabilities, security levels, connectivities, etc. In IIoT, device/service/user authentication is one of the key security issues and that is significantly varied as the applications [14]. For example, in the e-commercial system, digital certificates are widely used to guarantee the trustworthiness of an IIoT device, which can provide secure access control for IoT devices. In recent, the emerging decentralized ledger technologies (such as blockchain) have been used in IIoT to enhance

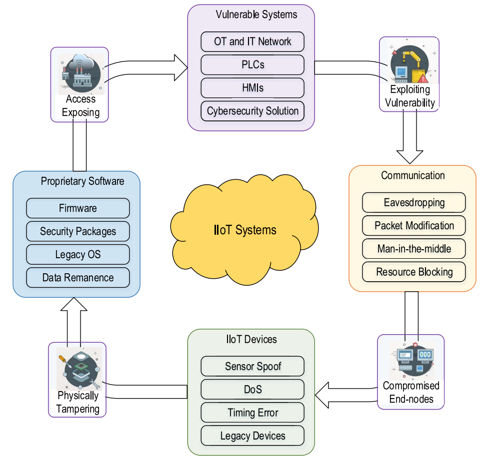


Fig. 1. Attack surfaces of IIoT.

the PKI and ensure digital cert management [1] by providing good auditability. In IIoT, two-factor authentication is sufficient. Since the devices are very different and the authentication schemes must consider the variation of protocols and standards. In the past few years, many research efforts have been conducted on authentication solution for applications in IIoT, such as healthcare, social networks, personalized privacy, cloud privacy, etc.

The IIoT must be able to provide industrial facilities and systems, including ICS, cyber–physical systems (CPS), industrial automation and control system (IACS), etc., with the ability to defend against new cyberthreats that take advantage of weaknesses and other attack vectors that come with the adoption of new technology [14]–[16]. To facilitate modern security architectures into existing industry systems, the following fundamental questions must be addressed: 1) to address the security architecture for existing systems and 2) to extend the modeled architecture artifacts to include security. To make it happen, the new systems must consider the business process, technologies, system architectures, and integration with existing cyber systems. The number of IIoT devices is increasingly growing, which presents new threats, vulnerabilities, and attack surface is significantly expanded. As shown in Fig. 1, the increased connectivity in IIoT also increases the attack surface for the software, access control, and critical processes.

Personal privacy has been well defined in information systems [6], [17], however, in an industrial environment (IIoT or Industry 4.0), privacy is still an open question. As discussed above, the smartization of devices, products, and operation technologies make the industry systems are exposing the maximum information the world has ever seen. Specifically, in the past few years, the information leaking incidents, such as `Miral`, `DDoS`, and `Stuxnet`, have made it very concern about the data privacy and personal data privacy in the industry environment. However, the IIoT faces difficulties from the following aspects: 1) it is unclear for data privacy in IIoT applications; 2) no privacy standardization for IIoT applications; 3) consent gathering from the user is inefficient; and 4) industry facilities profiling is very difficult to monitor. In this work, we summarize that the privacy issues in IIoT and its

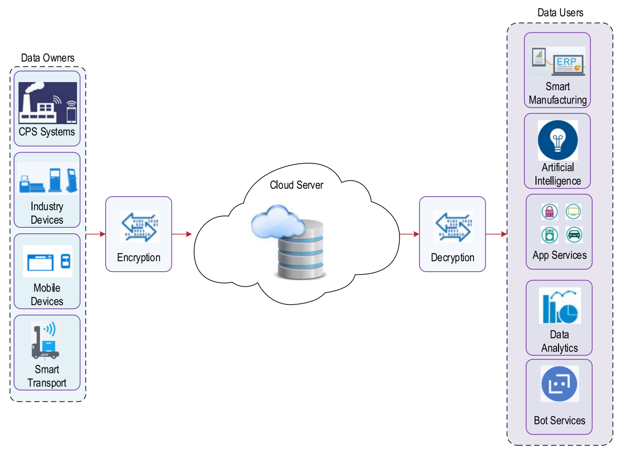


Fig. 2. Data encryption/decryption in IIoT.

application areas (such as real-time facial recognition systems and healthcare systems) cover:

1) the raw data storage;
2) insecure devices, applications, and users;
3) side-channel attack for IoT devices;
4) data encryption over resource-constrained devices;
5) insufficient security standardization;
6) diversity of IoT devices;
7) insecure data collection and sharing protocols.

In recent, a number of research works have been done on lightweight privacy-preserving, including lightweight credentials (domain credentials and generic credentials) management, authentication, multiuser authentication, and privacy-preserving solutions. Fig. 2 shows the data flow between data owners and data users, in which the cloud server(s) always provided by an untrusted third party.

In the past few years, many research efforts have been done on HE [13], [18], [19]. Most of these works focused on the computational data outsourcing to insecure cloud services [18]. The increasing use of IoT technologies in the industry makes it is important to leverage the security, privacy, and computing power. The computational expensive traditional data encryption way cannot match the needs of security and privacy in the Industry 4.0 environment. The HE allows computation over encrypted data, including multiplication (mul), addition (add), and constant-multiplication (cmul), when decrypted, matches the result of the operations that performed over plaintext. This features can hide the queries from the cloud servers and can further improve the privacy protection between cloud servers and data users. Lu [2] developed a communication-efficient privacy-preserving solution for IoT applications using BGN HE techniques. Meanwhile, researchers investigated the privacy preserving for IIoT devices from the access linkability for roaming service aiming at providing multilevel privacy preservation [20].

To solve this problem, a number of new lightweight homomorphic schemes for IoT were developed. Lu [2] developed a communication-efficient secure query scheme in a fog environment, in which both the data user (e.g., application) and data owner (e.g., an IoT device) can be privacy preserved using HE. Fiore *et al.* [18] developed a multikey homomorphic authenticator from the viewpoint of data outsourcing. The HE can make ciphertexts the same size after operations, which is suitable for some resource-constrained devices. HE contains addition and multiplication operations. Gentry proposed the full HE (FHE) by looking for an encryption scheme with low decryption complexity. However, the FHE still has some limitations: 1) the bootstrapping is computational expensive and cannot be performed by resource-constrained IIoT devices; 2) the size of ciphertext is too big that needs lots of storage space; and 3) the size of the public key is too big as well. FHE has been widely used in functional encryption, verifiable computing, secure multiparty computation, and attribute-based encryption (ABE).

In [21], a policy-based management system was developed to manipulate devices in the fog computing environment, which considers the user requests of resources, supports, etc. Jana *et al.* [5] developed an on-device sensor abstract that can avoid data leakage from accessing the raw data by data users. However, it is unable to deal with multiple data sources of data (multiple IoT devices) in the IoT environment.

However, most existing privacy-preserving solutions still face challenges, such as unclear data privacy, expensive computational costs, inefficiencies, etc. In the next section, we will introduce a new privacy-preserving scheme by refining the data privacy in IIoT.

## III. Privacy-Preserving IIoT System

It is reported that more than 90% of organizations dependent upon OT (such as those in the manufacturing, healthcare, and transportation industries) experiences at least one major cyberattack in the past two years [22], [23]. As discussed above, most IIoT system contains a large amount of facilities, locations, machines exchanging data through cloud platforms, and various applications. It is really difficult to map the complete attack surface of IIoT, Fig. 1 summarizes the critical attacks in IIoT systems. It can be seen that the surface areas include IIoT devices, vulnerable systems, proprietary software, and communication protocols. This work focuses on the privacy preserving in an IIoT system, which covers a `data owner`, a `cloud server` (or applications), and a `data user` (application).

### A. System Model

In our IIoT system, we consider three roles as shown in Fig. 5, which consists of three main entities.

*1) Data Owner (DO):* It can be a set of IoT devices $D = \{D_1, D_2, \ldots, D_n\}$ that can generate and process data with limited resources in terms of computation, speed, storage space, and memory size.

*2) Cloud Server:* The untrustworthy cloud servers can be a third-party hosting server that can store data generated by IoT devices; meanwhile, the cloud server can also offer computing services for both `data owners` and `data users`.

*3) Data User (DU):* It can be the IoT applications that use the data stored over the cloud server; it could be the data owner, e.g., an IoT device $D_i$ could be both `data owner` and `data user`. In this work, we use `app` to denote the data user.

*4) Sensitive Data:* A large part of data in IIoT must be protected against cyberattacks, including personal data, individual data, and operational data that created by IoT devices. Sensitive data needs particular security encryption solutions.

*5) Privacy Policies and Enforcement:* The privacy policies contain the rules that determine the authorized operations on sensitive data, which featured on *purpose, visibility, granularity*, and *obligations* of data [24]. Meanwhile, the policy enforcement processes are the usages of policies to protect associating data, access request, evaluation of policies, and the control access of data.

*6) Privacy Threats in IIoT:* Mainly include the identifiability threat, linkability, unauthorized data disclosure, excessive data disclosure, and profiling to infer interests and habits of individuals from their data and metadata.

### *B. Privacy Model*

For an IIoT with $n$ nodes, we have device set as $D = \{D_1, D_2, \ldots, D_n\}$. Each node $D_i$ can generate data $\mathbf{x}_i$. For privacy reason, node $D_i$ will not leak individual data $\mathbf{x}_i$ to others. For the cloud server, it should not know the $\mathbf{x}_i\ \forall i \in \{1, n\}$. To solve this problem, a straightforward way is to use encryption, to guarantee the privacy of data user, we will use HE $\mathbf{c}_i = \mathsf{Enc}(\mathbf{x}_i)$ here $\mathbf{c}_i$ will be transmitted to the cloud server by node $D_i$.

*Definition 1 [Data Owner Privacy (DOP)]:* For all IoT devices, each of them does not learn anything from other IoT devices, e.g., $D_i$ generated data $\mathbf{x}_i$ and $D_j$ generated data $\mathbf{x}_j$, $D_i$ can only transmit encrypted data $\mathbf{c}_i = \mathsf{Enc}(\mathbf{x}_i)$ and $D_i$ cannot learn anything from $\mathbf{c}_j$. Device $D_i$ might unable to aware if $D_j$ is active or not. Attacks from an external adversary are beyond the scope of this article. In this case, we say $D_i$ is privacy preserving for $D_j$. DOP means that no information is leaked about device's set elements to a malicious server, except the upper bound on the device's set size.

*Definition 2 [Data User Privacy (DUP)]:* In IIoT environment, a data user $\mathbf{U}$ might be an IoT device, or an application from the user. In this work, we assume that the application (app) is the data user that can conduct a query to the cloud server $\mathbf{S}$. From the viewpoint, a data user $U_k$ might use data from one or more data owners. Different data users do not have learn anything from each other. For example, $U_k$ does not have any connection with $U_\tau$ even they might consume the data from the same data owner $D_i$. In some scenarios, the data user also includes extra user privacy, such as preferences, location, etc.

*Definition 3 (Cloud Server Privacy):* For a cloud server $\mathbf{S}$, it might not be trustworthy for both DOs and DUs. $\mathbf{S}$ does not conduct data encryption or data decryption, so it can only be aware of who and when upload or download encrypted data but unable to know the contents of the data. In this work, $\mathbf{S}$ can also offer computing services for the queries from $\mathbf{U}$. The cloud server privacy preserving aims to prevent the data bleaches, such as insider theft, malware and ransomware, DDoS, etc., or misuse, disclose, modify, deny access, etc. If the IoT node $D_i$ learns no information (except the upper bound on size) about the subset of elements on the server that are NOT in the intersection of their respective sets.

*Definition 4 (Label Privacy):* At a data owner node, each piece of encrypted data is associated with a unique label $\ell$ (e.g., timestamp, ID, etc.). The class labels $\ell_{i,j}|\{j \in [1, M+i]\}$ may associate with the participated devices. In this work, we assume that the data users are happy to share the labeled data and do not consider the label privacy.

*Definition 5 [Differential Privacy (DP)]:* The DP aimed at protecting the privacy against learning by statistical queries on a database. For a randomized algorithm $\mathrm{Alg} : \mathbf{x} \to \mathbf{o}$ gives $\epsilon - DP$ if for all adjacent datasets $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{X}$ differing on at most one element. If the algorithm Alg satisfies (1), the algorithm Alg will satisfy $\epsilon - DP$ protection

$$P_r\big[\mathrm{Alg}(\mathbf{x}) \in \mathbf{o}\big] \leq \exp(\varepsilon) \times P_r\big[\mathrm{Alg}\big(\mathbf{x}'\big) \in \mathbf{o}'\big] \tag{1}$$

in which $P_r[\cdot]$ denotes the randomness of Alg on the data $\mathbf{x}$ and $\mathbf{x}'$.

In this work, the third-party cloud servers can only provide on-demand self-service. Gentry *et al.* [25] and Freedman *et al.* [26] proposed a single-server private information retrieval (PIR) protocol using keywords based on additive HE. In this work, we use the same interpolation polynomial. The communication per keywords is $O(\sigma \log |X| + \ell)$ and the size of the entire database is about $|\mathbf{X}| \cdot \ell$ [27]. It also works for multiple keywords handling and can also be used for resource-constrained IIoT device.

## IV. Proposed Scheme

As discussed above, in the past few years, a number of HE and its variants (such as FHE [27], SHE [13], and additively HE [28]) have been developed in verifiable computation outsourcing in cloud-based systems. This work aims at providing the data users different level of privacy preserving over data and protect data collected by the data owners from the following four aspects.

1) Provide settings that allow the data users to disable access to sensitive information.
2) At the cloud server, use the strongest data protection level for app. Use transport security when sending user or device data over the network.
3) The access for encrypted data over the cloud server must be authorized by both the data user and the cloud server.
4) Use the minimum amount of data required.

The conventional protocols, such as constrained application protocol (CoAP) [29], can provide IoT applications with secure communication, suitability, scalability, and privacy preservation. In typical IoT scenarios as shown in Fig. 2, the IoT mediators, including `IoT gateway, router, Internet,` and `servers` can mainly provide by the third party that cannot guarantee the security. In this work, we use the server to present IoT mediators. Meanwhile, the resource limitations of IoT devices might restrict the security solutions over `IoT devices`. To address the privacy preservation in IIoT, we need to clear the following major privacy problems.

1) Privacy awareness, which provides `data users` with discovery of services' privacy properties; this concerns

how IoT services might open communication channel between `app` and `devices`; `Data users` are not always aware when a device (`data owner`) presents and collects data. A data user can be informed to the presence of the data owners (e.g., PIR *sensor* $\leftrightarrow$ *cloud servers* $\leftrightarrow$ *app*); the `data user` does not have to clear if the `data owners` are alive or not.

2) Privacy preferences, for the cloud server, it might be able to learn from the data owner what data the device is collecting and what inferences might be possible; meanwhile, by analyzing the queries from the data users, it can learn what does the data user actually care about.

3) Privacy notification, IoT devices may not willing to interact to an IoT service in the IIoT environment, e.g., a temperature sensor might do not have to interact with an app (`data user`); however, for some specific services, e.g., product tracking in smart manufacture, the data user wants to be alerted by the location sensor.

In IIoT, a good security solution can ensure data is secure both in transit and at rest, including the confidentiality, integrity, and availability of data. From the viewpoint of privacy, good privacy means appropriate collection and use of information; the data collection/usage are transparent for data owners, cloud server, and data applications, respecting the rights and choose of individuals. This work aims at addressing the following key challenges.

### A. Data Preprocessing

Placing obligations and restrictions on the collection and use of "individual data." The raw form of data at each IIoT device is the participant's privacy to be protected from both data users and third-party cloud server. The data links between two IIoT devices can be used to measure the privacy degree

$$y = \mathcal{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t) \tag{2}$$

in which $\mathbf{x}_i$ denotes the data at node $D_i$. A labeled program $\mathcal{P}$ is a tuple $(f, \ell_1, \ldots, \ell_n)$, and $f$ is an HE allowed function of $n$ variables and $\ell_i \in \{0, 1\}^*$ is a label for the $i$th input of $f$.

In this work, we introduce a lightweight data transmission scheme that includes the following four algorithms: $\mathsf{KeyGen}()$, $\mathsf{Enc}()$, $\mathsf{Eval}()$, and $\mathsf{Dec}()$.

*1) Key Generation Algorithm:* This stage uses the $\mathsf{keyGen}(1^\lambda)$ algorithm to create keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk})$, in which $\mathsf{pk}$ denotes a public key, $\mathsf{sk}$ denotes a secret authentication key, and $\mathsf{vk}$ denotes a verification key, respectively.

*2) Encryption Algorithm:* The $\mathsf{Enc}()$ algorithm creates ciphertext $\mathsf{c}$, as $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x})$, in which $\mathbf{x}$ is the dataset, and $\ell$ is the label of $\mathbf{x}$.

*3) Evaluation Algorithm:* At the cloud server, when received a request for using data from an `app` (data user), the server will invoke the $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{c})$ to create a new ciphertext based on the request $\mathcal{P}$.

*4) Decryption Algorithm:* Using the $\mathsf{sk}$, with $\mathsf{Dec}(\mathsf{sk}, \mathcal{P}, \mathsf{c})$, the data user is able to verify and decrypt the message and label $\ell$ and decide to reject or accept the data retrieved from the server.

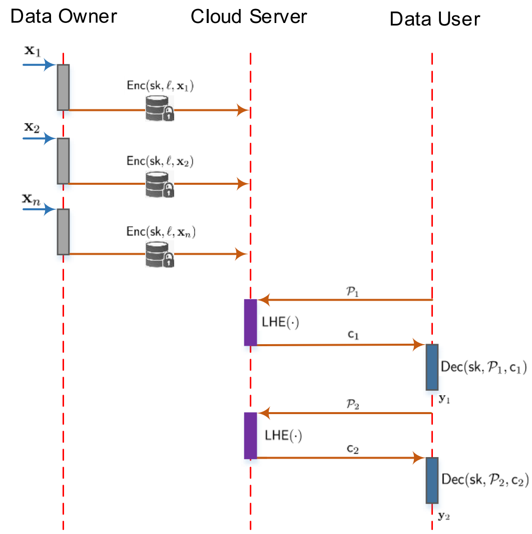


Fig. 3. Single-user data enc.

### B. Lightweight Privacy-Preserving Protocol

In IIoT, many applications (apps) may take computation tasks over encrypted data and looking for statistically relevant features across the encrypted data. Typically, such tasks are carried by computational powerful devices and each data query takes computational expensive algorithms, such as $\mathsf{Enc}$, $\mathsf{Eval}$, $\mathsf{Dec}$, etc., which stops the resource-constrained IIoT devices to perform this tasks in many applications, such as ehealthcare, smart home, etc. The proposed solution can shift the computation from IIoT devices to the untrusted cloud server and offers a tradeoff between privacy protection in terms of *data owner, cloud server, data user*, and the computational efficiency.

Assume we have data space $\mathbf{x} \in \mathcal{X}$, label space $\ell \in \mathcal{L} \subset \{0, 1\}$, and an admissible circuits $\mathcal{F}$, including multiplication, addition, and constant-multiplication [12].

*1) Single-User Protocol:* In a cloud system, for a data owner $\mathbf{A}$. The system generates keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})$ using $\mathsf{KeyGen}(1^\lambda)$. As shown in Fig. 3, a data owner takes as input as the $\mathsf{sk}$, a label $\ell \in \mathcal{L}$, and a message $\mathbf{x}_i \in \mathcal{X}$ to encrypt message using $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x})$ and outputs a ciphertext $\mathsf{c}$, which will be uploaded to the cloud server $\mathbf{S}$. When a data user wants to use the data, it will sent a request of allowed arithmetic circuit $\mathsf{f}$. The $\mathsf{f}$ takes as input the $\mathsf{ek}$, ciphertexts $\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_t$, the cloud server will perform the evaluation algorithm $\mathsf{Eval}(\mathsf{ek}, f, \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_t)$ and return a ciphertext $\mathsf{c}$ to data user.

The data user takes as input the $\mathsf{sk}$, a label program $\mathcal{P} = (f, \ell_1, \ell_2, \ldots, \ell_t)$, and the ciphertext $\mathsf{c}$, using decryption function $\mathsf{Dec}(\mathsf{sk}, \mathcal{P}, \mathsf{c})$ can output a message $f(\mathbf{x}')$, as

$$\Pr\left[\mathbf{x}'\right] = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t) \tag{3}$$

is negligibly close to 1, then the solution is said to correctly evaluate an $f \in \mathcal{F}$ for all keys.

In the IIoT environment, a cloud server is more powerful than data owner and data users regarding computing capabilities and storage capacity. However, for an IIoT node, it has to run $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_i)$ when each time the node creates

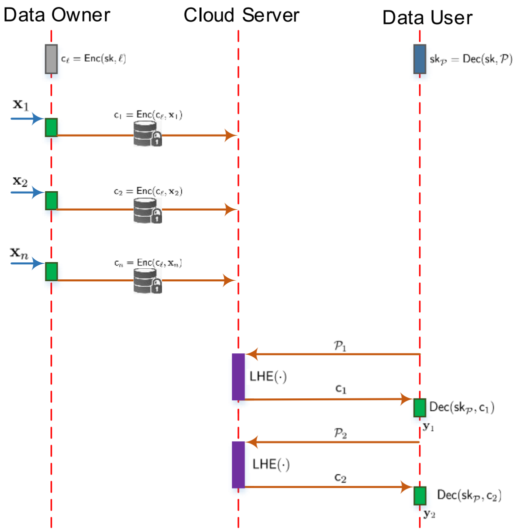


Fig. 4. Multiple user data enc.

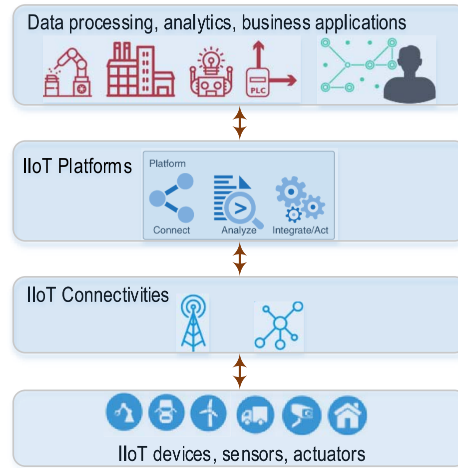


Fig. 5. IIoT infrastructure.

a message $\mathbf{x}_i$, which is computation expensive for resource-constrained nodes. Meanwhile, for the data user (always not powerful devices) it needs to run computational expensive $\mathsf{Dec}(\mathsf{sk}, \mathcal{P}, c)$ for each data query.

A straightforward idea is to outsource the computation at data owner and data users to the cloud servers. Using the labeled HE scheme [12], it is possible to reduce computation of $\mathsf{Enc}(\cdot)$ and $\mathsf{Dec}(\cdot)$, as shown in Fig. 4. It can not only provide lightweight data encryption but also match the privacy requirements addressed in Section III.

*2) Computation Split at Data Owners and Data Users:* It is noted that the computation cost at data owner comes from $\mathsf{Enc}(\cdot)$, for each piece of data $\mathbf{x}_i$, the data owner has to perform $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_i)$. Since the $(\mathsf{sk}, \ell)$ are always the same, a straightforward way is to divide the $\mathsf{Enc}(\cdot)$ into two parts

$$\mathsf{c}_\ell = \mathsf{Enc}(\mathsf{sk}, \ell) \tag{4}$$
$$\mathsf{c}_i = \mathsf{Enc}(\mathsf{c}_\ell, \mathbf{x}_i). \tag{5}$$

Equation (4) takes as input a label and the secret key and output ciphertext $\mathsf{c}_\ell$, which can be reused by messages created by the same node. Both (4) and (5) are homomorphically correct in the sense that $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_i)$ [12]. As shown in Fig. 4, a data user must be able to correct decrypt the ciphers. Similarly, at a data user, given $\mathsf{sk}$ and an allowed circuits, the decryption algorithm can be divided into two parts

$$\mathsf{sk}_\mathcal{P} = \mathsf{Dec}(\mathsf{sk}, \mathcal{P}) \tag{6}$$
$$\mathbf{x}'_i = \mathsf{Dec}(\mathsf{sk}_\mathcal{P}, \mathsf{c}). \tag{7}$$

Here both (6) and (7) are homomorphically correct in the sense that $\mathsf{Dec}(\mathsf{sk}, \mathcal{P}, \mathsf{c})$ [12]. This splits the computation into a fixed function and a dynamic function with dynamic computation complex.

In an IIoT environment with multiple IIoT devices, the key generation algorithm will generate a master public key and a master secret key $\mathsf{KeyGen}(1^\lambda) \rightarrow (\mathsf{msk}, \mathsf{mpk}, \mathsf{uek})$. The master public key will be used to generate a public key associated with its encrypted data. Each data owner can encrypt using its public keys and user's master public key $(\mathsf{mpk})$ to guarantee the data encrypted by a data user cannot be decrypted by another data user. Decryption requires the $\mathsf{msk}$ along with $\mathsf{pk}$ of data owner/users involved.

For an IIoT node $D_i$, the encryption algorithm $\mathsf{Enc}(\mathsf{mpk}, \mathsf{usk}, \ell_i, \mathbf{x}_i)$ outputs a ciphertext $\mathsf{c}_i$. At server $\mathbf{S}$, evaluation algorithm will perform $\mathsf{Eval}(\mathsf{mpk}, f, \mathsf{c}_1, \ldots, \mathsf{c}_t)$ and returns a ciphertext $\mathsf{c}_s$. A data user $D_j$, for example, can use $\mathsf{Dec}(\mathsf{sk}, \mathsf{upk}, \mathcal{P}, \mathsf{c}_s)$ returns a message $\mathbf{x}'$. For all honestly generated keys $(\mathsf{mpk}, \mathsf{msk})$, all data user key pairs $(\mathsf{upk}_i, usk_i), \ldots, (\mathsf{upk}_t, \mathsf{usk}_t)$, if we have

$$\Pr\left[\mathbf{x}'\right] = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t) \tag{8}$$

is negligibly close to 1, then we can say the protocol is privacy secure for data owner, server, and data users.

*Correctness:* To guarantee the correctness of the above algorithms, the system applies $\mathbf{Eval}(\cdot)$ on data and label $\mathsf{c}_1, \ldots, \mathsf{c}_n$ corresponding to the label set $\ell$, respectively, the result will be a ciphertext $\mathsf{c}_{(f,\mathbf{x})}$ that verifies against function $f$, labels $\ell$, and message $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

*Theorem 1:* In the semi-honest model, the proposed protocol can solve the above privacy-preserving problems between multiple data owners and multiple data users, in which a data owner or data user can follow the lightweight protocol to keep privacy without data leaking.

*Proof: (Security):* suppose the protocol is insecure. Then there is a probabilistic polynomial time-real adversary $\mathcal{A}$ that does not have corresponding PPT ideal adversary $\mathcal{A}'$ exists that makes $(P_1(\mathbf{x}_1), \ldots, P_n(\mathbf{x}_n), \mathcal{A})$ and $(P'_1(\mathbf{x}_1), \ldots, P'_n(\mathbf{x}_n), \mathcal{A}')$ are computational indistinguishable. ■

Basically, an IIoT system consists of four main entities: 1) *intelligent assets*; 2) IIoT *infrastructure*; 3) *analytics* and *applications*; and 4) *users*. Fig. 5 shows an architecture of an IIoT system, in which the *intelligent assets* could be the IoT sensors, actuators, industry machineries, etc., that can generate and store data. The IIoT infrastructure provides connectivities between components in IIoT. Applications or devices that can consume data are data users.

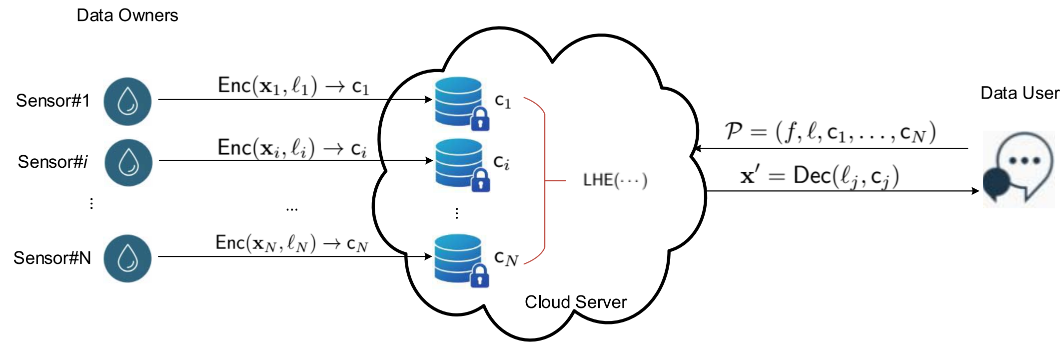


Fig. 6. Lightweight privacy data encryption protocol.

1) A smart sensor or actuator, for example, can generate, collect, process, transmit, or store data and an actuator is able to conduct physical functions, such as closing or opening a door when a command is conducted.
2) The IIoT infrastructure, includes IoT gateway and IIoT platforms, can provide connectivities between components in IIoT and allow data real-time analysis. The most popular IoT platforms include the Azure IoT hub, where device data is sent to the cloud.
3) Analytics and applications, it provides detailed analysis of remaining data and relies on the applications that an IIoT can provide, where data-intensive processing and analysis takes place.
4) IIoT entities include IIoT users, service providers, customers, etc.

## V. Evaluation and Discussion

The increasing popularity of IIoT shows great promises both new conveniences and new privacy concerns. In this section, we will introduce an air quality monitoring applications in Bristol. We examine an IIoT scenario as shown in Fig. 6 to test proposed solutions that allow the data owner and data users to protect themselves from smart home privacy vulnerabilities. The *sensors* are deployed at different sites as shown in Fig. 7 and Table I. The dataset collected by these sensors is provided by the Bristol council and this case studies we have performed motivated components of a general IIoT solution. To evaluate the proposed scheme, we use the air quality dataset to test the above data. The data are acquired from 15 sites in the Bristol area as shown in Table I. The location of these sites is highlighted in Fig. 7.

In the test, the data captured by sensors at *AURN St Pauls* were encrypted and uploaded to a third-party cloud server at UWE. We define these sensors are *data owner*, the `data user` is an app over mobile phone that can send query to the server. The raw data was labeled using the timestamp, the query $\mathcal{P} = \{f_1, f_2\}$, $f_1$ is a *const−multiplication* algorithm, and $f_2$ is an *addition*. When enquiring the air quality data at *AURN st Pauls*, the `data user` app sends a query $\mathcal{P} = (f_1, 05082019)$, in which $f_1$ is the cost-multiplication and constant was defined as 1, and $\ell = 05082019$ is the time-label. When send a query for asking the quality data of August 5, 2019.

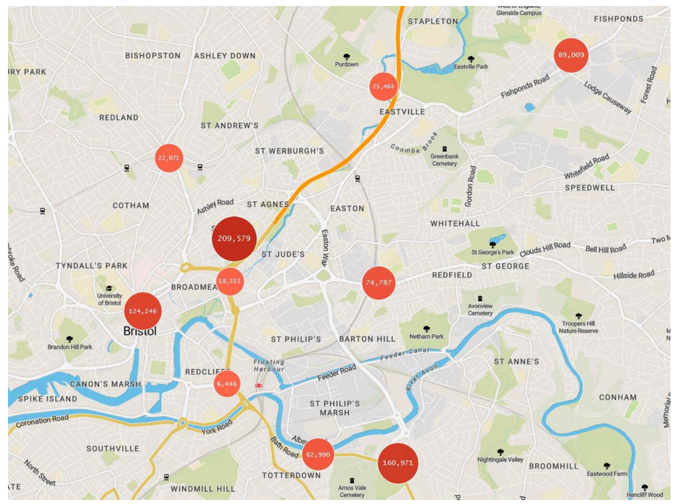


Fig. 7. Air condition acquisition sites in Bristol.

We tested the proposed scheme over a server with Ubuntu 18.04, Intel Core i7 7700K Quad Core Dedicated Server (4.2 GHz × 8), 32-GB RAM. In the test, the size of $\mathsf{sk}$, $\mathsf{pk}$, $\ell$, and *Nounce* is 128 bits. Privacy analysis: 1) the data owner $ID_{\text{AURN}}$ learns nothing from other data owners (such as $ID_{\text{Fishpond}}$, *etc*.); 2) the cloud server learns nothing from $ID_{\text{AURN}}$, and learns nothing from the data user as well; and 3) the data user learns nothing from other data users. Table III addresses the air quality data at *AURN St Pauls*.

The homomorphic addition operation takes 0.001 ms, and homomorphic mul operation takes 0.151 (ms). In the proposed solution, the $\mathsf{Enc}(\mathsf{sk}, \ell)$ takes 557.725 ms, $\mathsf{Enc}(\mathsf{c}_l, \mathbf{x})$ takes 0.055 (ms), at the data user app, the $\mathsf{Dec}(\mathsf{sk}, \mathcal{P})$ takes 6.083 ms, and the $\mathsf{Dec}(\mathsf{sk}_{\mathcal{P}}, \mathsf{c})$ takes 2.685 ms, it is clear that the proposed scheme can significantly reduce the computation costs at both data owner and data user. Table II compares the time consuming of each computing component in the proposed scheme when the key size and label size are different.

Each site first generates keys using $\mathsf{KeyGen}() \to (\mathsf{ek}, \mathsf{sk}, \mathsf{vk})$, and then share $\mathsf{sk}$ and $\mathsf{pk}$ with the server. Each node then encrypts the data using $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x})$ to generate labeled ciphers and transmitted to the cloud server. In this work, each data owner submitted its data to an untrusted cloud server and the server allocate it a label, $\ell_{\text{BLD}}$, the $\mathsf{ID}$ of BLD is 203. The cloud server with $\mathsf{Eval}$ the data before doing further processing. The data user decrypted results are exactly

TABLE I
AIR QUALITY DATA CONTINUOUS ($ug/m^3$)

| ID | Data Onwer | Records | $NOx$ | $NO_2$ | $NO$ | PM10 | PM2.5 |
|---|---|---|---|---|---|---|---|
| 203 | Brislington Depot (BLD) | 160,970 | 74.9 | 30.0 | 29.2 | – | – |
| 215 | Parson Street School (PSS) | 152,188 | 116.2 | 32.3 | 54.1 | – | – |
| 270 | Wells Rd (WLR) | 140,021 | 161.7 | 49.5 | 73.1 | – | – |
| 206 | Rupert Str (RPS) | 113,951 | 280.3 | 90.0 | 120.8 | – | – |
| 452 | AURN St Pauls (ASP) | 113,171 | 55.8 | 5.9 | 32.6 | 20.3 | 6.0 |
| 375 | N. Road Station (NRS) | 96,407 | 65.0 | 29.8 | 23.0 | – | – |
| 463 | Fishponds Rd (FPR) | 91,347 | 65.9 | 38.3 | 18.0 | – | – |
| 395 | Shiner's Garage (SSG) | 74,787 | 174.8 | 73.3 | 101.8 | – | – |
| 447 | Bath Rd (BAR) | 62,990 | 91.5 | 47.3 | 44.3 | – | – |
| 271 | Trailer Portway P (TPP) | 43,824 | 61.5 | 35.7 | 16.8 | – | – |
| 209 | IKEA M32 (IKM) | 25,464 | 159.0 | 77.6 | 51.1 | – | – |
| 459 | Cheltenam Rd (CTR) | 22,071 | 72.5 | 38.3 | 22.3 | – | – |
| 500 | Temple Way (TPW) | 18,552 | 16.8 | 12.0 | 3.1 | 14.5 | – |
| 501 | Colston Avenue (CSA) | 10,294 | 52.9 | 25.8 | 17.7 | 30.8 | – |
| 228 | Temple Meads St. (TMS) | 6,446 | 1,174.3 | 122.0 | 680.3 | – | – |

TABLE II
TIMINGS OF THE DATA QUALITY QUERIES

| key size (bits) | label size | Operations | Times |
|---|---|---|---|
| 128 | 1024 | $\mathsf{Enc}(\mathsf{sk}, \ell)$ | $150.7ms$ |
| 128 | 1024 | $\mathsf{Enc}(\mathsf{c}_l, \mathsf{x})$ | $0.06ms$ |
| 128 | 1024 | $dec(\mathsf{sk}, \mathcal{P})$ | $2.204ms$ |
| 128 | 1024 | $\mathsf{Dec}(\mathsf{sk}_{\mathcal{P}}, \mathsf{c})$ | $0.723ms$ |
| 128 | 1024 | $\mathsf{add}$ | $0.001ms$ |
| 128 | 1024 | $\mathsf{mul}$ | $0.042ms$ |
| 128 | 2048 | $\mathsf{Enc}(\mathsf{sk}, \ell)$ | $557.725ms$ |
| 128 | 2048 | $\mathsf{Enc}(\mathsf{c}_l, \mathsf{x})$ | $0.055ms$ |
| 128 | 2048 | $dec(\mathsf{sk}, \mathcal{P})$ | $6.083ms$ |
| 128 | 2048 | $\mathsf{Dec}(\mathsf{sk}_{\mathcal{P}}, \mathsf{c})$ | $2.685ms$ |
| 128 | 2048 | $\mathsf{add}$ | $0.001ms$ |
| 128 | 2048 | $\mathsf{mul}$ | $0.151ms$ |

TABLE III
AIR QUALITY DATA AT ASP ON AUGUST 5, 2019 ($ug/m^3$)

| ID | Data Onwer | $NOx$ | $NO_2$ | $NO$ | PM10 | PM2.5 |
|---|---|---|---|---|---|---|
| 452 | AURN St Pauls (ASP) | 59.9 | 10.3 | 32.3 | 15.5 | 7.0 |

the same with the original data at the data owner, as shown in Table III. The result shows as $NOx = 59.9$, $NO2 = 10.3$, $NO = 32.3$, $PM10 = 15.5$, and $PM2.5 = 7.0$.

Furthermore, when an application needs to access the data, e.g., someone wants to know the average temperature in Fishponds Rd in the past five years (from 2014 to 2019), he may send a function $f_{\text{avg}}$ to the server together with the label $\ell_{\text{FPR}}$, the server can perform a homomorphical and return the results.

In practice, IoT devices and applications create or collect a huge amount of data, which is a valuable business asset. Typically, the data owner is the device or app that generated the data itself. In some cases, the data are aggregated before being encrypted and sent to the cloud server. New classes of IoT devices that can collect information that was not previously available are emerging, such as *wearable, smart sensors, etc.*, individual data in industry networks are facing increasing sophisticated cyberattacks. As a very modern development, the IIoT has to face new challenges using new approaches include lightweight cryptography, computational intelligence, and distributed ledger technology (DLT) techniques.

## VI. CONCLUSION

In this article, we presented the privacy between data owners, third-party cloud server, and data users. A lightweight privacy-preserving protocol for the data owner–server–data user model is proposed based on the labeled HE. The evaluation computation costs are shifted from resource-constrained IoT devices to the third-party powerful server without losing security and privacy strength. It is an efficient and practical scheme in the IIoT environment and allows a remote, nonconfident, cloud computing to perform complex computational over encrypted data, and permits the data owner, data users to verify the exactitude of decryption. We further propose effective protocols that can be easily incorporated with existing IIoT and so that can significantly reduce the computational costs.

## REFERENCES

[1] K.-K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial Internet-of-Things: Research challenges and opportunities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3567–3569, Aug. 2018.

[2] R. Lu, "A new communication-efficient privacy-preserving range query scheme in fog-enhanced IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2497–2505, Apr. 2019.

[3] J. Zhang, Z. Wang, L. Shang, D. Lu, and J. Ma, "BTNC: A blockchain based trusted network connection protocol in IoT," *J. Parallel Distrib. Comput.*, vol. 143, pp. 1–16, Sep. 2020.

[4] L. Zhou, K.-K. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the Industrial Internet of Things: An overview of approaches to safeguarding endpoints," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 76–87, Sep. 2018.

[5] S. Jana *et al.*, "Enabling fine-grained permissions for augmented reality applications with recognizers," *Presented at the 22nd USENIX Security Symp. (USENIX Security)*, Washington, DC, USA, 2013, pp. 415–430.

[6] K. Fan, W. Jiang, H. Li, and Y. Yang, "Lightweight RFID protocol for medical privacy protection in IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1656–1665, Apr. 2018.

[7] R. Yugha and S. Chithra, "A survey on technologies and security protocols: Reference for future generation IoT," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102763.

[8] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in iot-enabled smart metering systems," *Future Gener. Comput. Syst.*, vol. 78, pp. 547–557, Jan. 2018.

[9] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3590–3598, Aug. 2018.

[10] S. N. Foley, D. Gollmann, and E. Snekkenes, "Computer Security—ESORICS 2017," *Proc. 22nd Eur. Symp. Res. Comput. Security II*, Oslo, Norway, Sep. 2017.

[11] J. H. Cheon and T. Takagi, *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, vol. 10032. Heidelberg, Germany: Springer, 2016,

[12] M. Barbosa, D. Catalano, and D. Fiore, "Labeled homomorphic encryption—Scalable and privacy-preserving processing of outsourced data," IACR Cryptol. ePrint Archive, Lyon, France, Rep. 2017/326, 2017.

[13] J. H. Cheon and J. Kim, "A hybrid scheme of public-key encryption and somewhat homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1052–1063, 2015.

[14] Z. Ma, A. Hudic, A. Shaaban, and S. Plosz, "Security viewpoint in a reference architecture model for cyber-physical production systems," in *Proc. IEEE Eur. Symp. Security Privacy Workshops (EuroS PW)*, Paris, France, Apr. 2017, pp. 153–159.

[15] H. Mahdikhani, R. Lu, Y. Zheng, J. Shao, and A. A. Ghorbani, "Achieving $o(log^3 n)$ communication-efficient privacy-preserving range query in fog-based IoT," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5220–5232, Jun. 2020.

[16] M. Alloghani *et al.*, "A systematic review on the status and progress of homomorphic encryption technologies," *J. Inf. Security Appl.*, vol. 48, Oct. 2019, Art. no. 102362.

[17] M. M. H. Onik, C. Kim, and J. Yang, "Personal data privacy challenges of the fourth industrial revolution," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, PyeongChang, South Korea, Feb. 2019, pp. 635–638.

[18] D. Fiore, A. Mitrokotsa, L. Nizzardo, and E. Pagnin, "Multi-key homomorphic authenticators," in *Proc. II, 22nd Int. Conf. Adv. Cryptol. (ASIACRYPT)*, vol. 10032, 2016, pp. 499–530.

[19] M. Beunardeau, A. Connolly, R. Geraud, and D. Naccache, "Fully homomorphic encryption: Computations with a blindfold," *IEEE Security Privacy*, vol. 14, no. 1, pp. 63–67, Jan./Feb. 2016.

[20] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 46–57, Feb. 2014.

[21] C. Dsouza, G.-J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IEEE IRI)*, Redwood City, CA, USA, Aug. 2014, pp. 16–23.

[22] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Syst. J.*, vol. 14, no. 1, pp. 310–320, Mar. 2020.

[23] Y. Lu, J. Li, and Y. Zhang, "Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted IIoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2553–2562, Apr. 2020.

[24] A. Al-Hasnawi, I. Mohammed, and A. Al-Gburi, "Performance evaluation of the policy enforcement fog module for protecting privacy of IoT data," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, Rochester, MI, USA, May 2018, pp. 0951–0957.

[25] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology (CRYPTO)*, Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds. Heidelberg, Germany: Springer, 2013, vol. 8042, pp. 75–92.

[26] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Proc. Theory Cryptogr. Conf.*, 2005, pp. 303–324.

[27] H. Chen, Z. Huang, K. Laine, and P. Rindal, "Labeled PSI from fully homomorphic encryption with malicious security," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 1223–1237.

[28] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 1333–1345, 2018.

[29] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2299–2308, Apr. 2019.